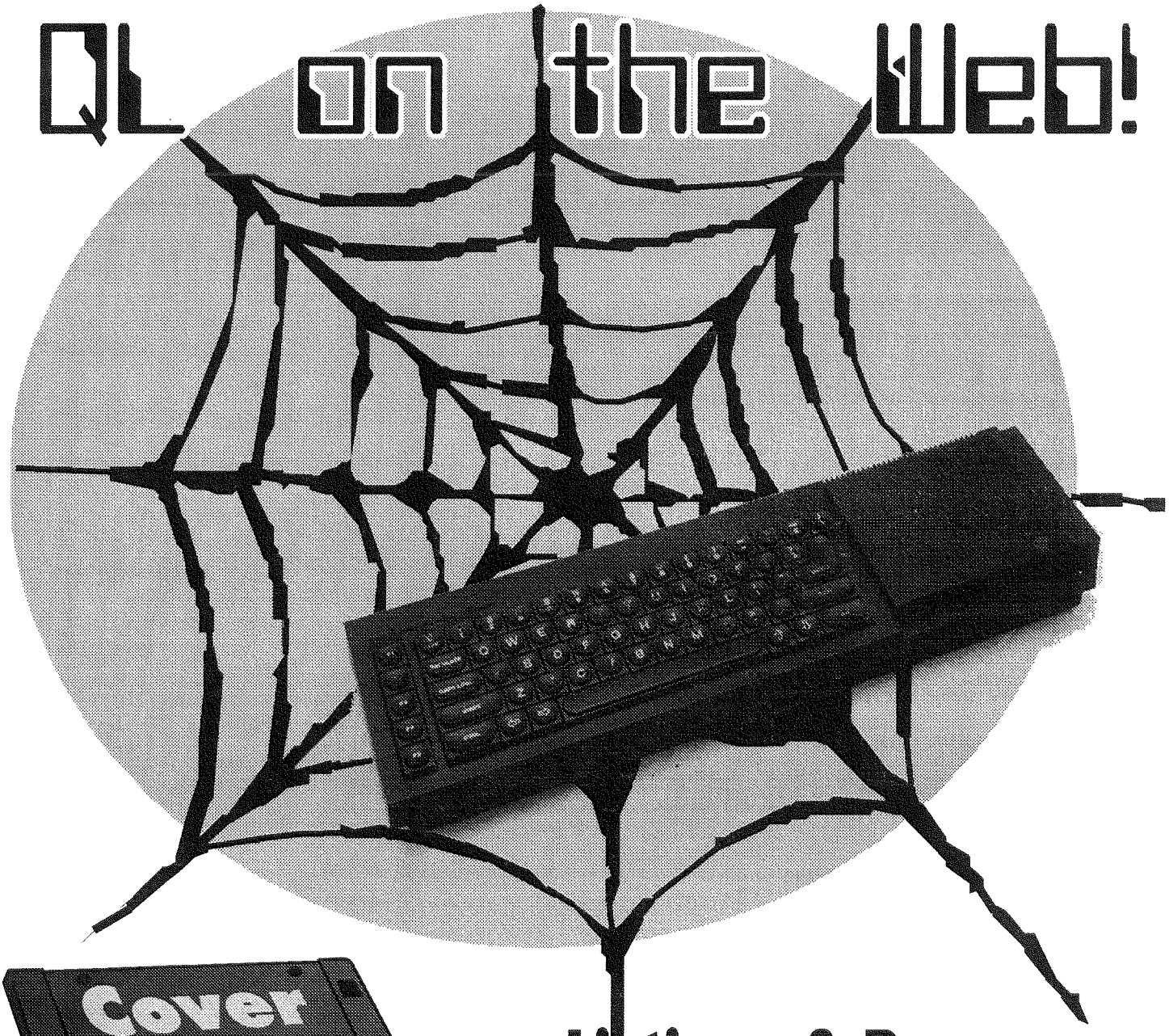
QL Today

Volume 4
Issue 6
March/April
2000

ISSN 1432-5454

The Magazine about QL, QDOS, Sinclair Computers, SMSQ...

QL on the Web!



Listings & Programs of Volume 4 QL MidiPlayer 2 & RWAP software demos!

Contents

3 Editorial
4 News
7 Cover Disk Notes
8 2000 Hove QUANTA Workshop *John Hall*
10 The QL Family Tree - The 90's *Dilwyn Jones*
14 Of Webs Echoes Leaps and Surfing Black Boxes *J. Dent*
18 Q 40 Tips *Roy Wood*
18 Image Viewer PQIV *Claus Graf*
19 Jpegs how to get some - or a review of a Kodak 215 zoom digital camera *Bill Waugh*
20 Archivers Control Panel *Roy Wood*
24 MonoCall *Nick Cheesman*
26 The function of PROCedures and the procedure for writing FuNctions Part 2: Why FuNctions? The result's the reason! *Mark Knight*
28 You and Your Software - Just good Friends? Part 7: Manuals, the great unread *Geoff Wicks*
30 CAUTION: HOTKEYS - Don't Burn Your Fingers, Part 3 *David Denham*
33 A dozen Progs You shouldn't live without *Darren Branagh*
34 QLTdis - part four *Norman Dunbar*
42 Gee Graphics! (On the QL?) - part 15 Lines and Planes in 3 space *Herb Schaaf*
44 Letter Box
48 MIDIPlayer 2 *Al Boehm*
50 Byts of Wood *Roy Wood*
54 Postcard: Screen Saver - You decide!
55 US Show 2000

Advertisers

in alphabetical order

Jochen Merz Software 36, 37
QBOX USA 49
QBranch 16, 17
Quanta 31
RWAP 51
W.N. Richardson (EEC) 25
TF Services 45
Geoff Wicks 41

QL Today

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software
Im stillen Winkel 12
47169 Duisburg
Germany

Tel. +49 203 502011
Fax +49 203 502012
Box1 +49 203 502013
Box2 +49 203 502014
email: JMerz@j-m-s.com

English office:

Q Branch
The Bank Volt
6 Coronation Buildings
Ham Road
Sussex BN11 2NN
United Kingdom

Tel. +44 1903 200212
Mobile +44 836 745501
Fax +44 1903 208070
Fax +44 1273 381577
email: qbranch@qbranch.demon.co.uk

Editor:

Dilwyn Jones
41 Bro Emrys
Tal-Y-Bont, Bangor
Gwynedd
United Kingdom LL57 3YT

Tel. +44 1248 354023
Fax +44 1248 354023
email: dilwyn.jones@dj.softnet.co.uk
email: QLToday@J-M-S.com

QL Today is published bi-monthly, our volume begins on 15th of May. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.
Payment in DM (drawn on a German bank) can be made by either Cheque or Euro-Cheque, Payment in £ (drawn on an English bank) can be made by Cheque. Cheques should be made payable to Jochen Merz Software (German office) or QBranch (English office).

Credit Card holders may subscribe by either calling or sending their Credit Card number and Expiry date. Credit cards will be charged in DM (German office) or in £ (English office).

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email or into one of the JMS-BBS's. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising deadlines are as follows:

Issue 1: 30 April	Issue 2: 30 June
Issue 3: 30 August	Issue 4: 30 October
Issue 5: 30 December	Issue 6: 28 February

Editorial

Dilwyn Jones

Several important developments for the QL scene are taking place concurrently at the moment. The main one is no doubt the "TCP/IP" project in Switzerland. Jon Dent has made a point of keeping QL Email Mailing List members updated on progress and he has kindly written an article for us on this subject so that you can see that it really is happening and quite well advanced too.

This in turn led Joachim van der Auwera to make ProWesS freely distributable. This can only be good - the fact that Prowess is free (although still Joachim's copyright) means that the improved windowing, scaleable fonts and HTML reader etc can offer a way forward for QL software such as Web browsers and Web page development tools in addition to bringing the concept of standard printer drivers closer for all. At first, I was worried this meant the end of ProWesS support and development, then he goes and proves me completely wrong by releasing a new version with improved Q40 support.

I am also aware of some software projects going on behind the scenes in relation to HTML page creation and so on - Geoff Wicks is building some support for HTML output into his QL2PC software and I have been working on a program called DOC2-HTML to allow basic Web pages to be created in Quill. Of course, we already have Jonathan Hudson's Lynx browser and email software which Jon Dent hopes to support with his software. I seem to recall that there is an ftp client software for QDOS too, although I cannot remember who the author is. This is not the first time QDOS has had access to the Web of course - uQLx users probably have that honour, since Jonathan Hudson made use of the underlying TCP/IP facilities of Linux via the uQLx emulator for his email software to gain access to the Web. If memory serves me right, there was also a version of the Mosaic browser for the QL at one time, though it never gained widespread use.

The "colour drivers" are continuing to develop though so far only for the Q40. Better like this in my opinion - get them working properly first and then get them to work on the other systems. I gather that Roy Wood has lent an Aurora system to Tony Tebby in an attempt to assist with development, and work should definitely proceed on the QXL versions and the way prepared for QPC to join the rank of systems eventually enjoying the new GD2 facilities!

Another cover disk with this issue I'm afraid. Rich Mellor has kindly supplied some demo versions of his software, Norman Dunbar's assembler series provides some material such as an updated GWASL assembler and the latest QLTdis code. Although we were not exactly bombarded with modules for CueDark as we had hoped, we did receive some interesting submissions which are on the disk - we hope you'll help us select a winning entry. Apologies for having to ZIP everything on the disk, we ran out of space and felt it was a good excuse to get everyone using Archiver's Control Panel (see Roy's article).

All this progress and what do I do? Generate an article looking back at the 80s and 90s for QLers. Still, it was an interesting exercice for me and brought back some memories - nostalgia isn't what it used to be, as they say.

Cartoon

NEWS

ProWesS News from PROGS

Joachim van der Auwera has decided to take ProWesS, the PROGS windowing system, off the market to help the QL move on, and to make it freely distributable. While PROGS retain copyright on the software, it can now be freely downloaded from their Web site and used by QLers. Given the move towards better comms facilities on the QL and that ProWesS includes an HTML reader for example, this is definitely good news and will encourage use of the advanced facilities of ProWesS. Programming information for it is also available from sources such as Joachim's Web site. ProWesS will be downloadable from the Web site

http://www.triathlon98.com/Joachim/

in time, but until then users can download the files from the Triathlon98 anonymous ftp server

ftp://triathlon98.com

including programming information.

On 13 February 2000 PROGS released a revised ProWesS version for Q40 users, including changed caching handling in the DLL manager to properly work on the Q40 (this requires SMSQ/E 2.94 or higher or compatible caching support); added a new mode 33 bitmap driver for the Q40 and support for mode 33 detection in the screen driver; startup file has undergone some minor changes (adding some extra pauses) to make loading work properly on the Q40.

Simon Goodwin

Congratulations to QL guru Simon Goodwin and partner Chris Lyle on the birth of their son Ingo Erin Lyle-Goodwin. Ingo was born on 15:06 on 22nd February 2000, weighing 7 pounds 6.5 ounces, after a long and rather difficult birth. Simon is in the process of moving house and has rather a lot on his plate at the moment. Any correspondence for Simon can be sent via his previous address in Oldbury and it will get redirected to his new home.

Illness

Our best wishes to Rich Mellor, who has been unwell of late, and also to Bill Richardson who has spent some time in hospital recently. Best wishes to both for a full and speedy recovery.

MINIVIEW

This is a new compact plain text file viewer by Dilwyn Jones, even smaller than the original text file viewer by the same author. The version without QLiberator compiler runtimes is only about 7K long!

It is freeware and can be used for distributing text files such as instructions for software, or just used as a compact and ultra simple to operate text file viewer. It simply lets you load a text file and scroll through it using the cursor keys. There is a PRINT command to allow the file to be printed.

If a filename is passed to the program as a command parameter, the program loads and displays that file. This allows it to be used with FileInfo for example. Otherwise, it prompts for a filename.

MiniView will soon be available from PD libraries and from his Web site:

http://www.soft.net.uk/dj/index.html

uQLx NEWS

The 28.1.00 release of the uQLx emulator from Richard Zidlicky has the following changes:

USE_VM for Linux x86 and SPARC Solaris. Allows 12-20% speed improvement, I wonder who needs that in the times of GHz CPUs? This is very stable on Linux x86 2.0.35, but crashes Linux m68k 2.2.{6,10} (thus disabled for this architecture by default).

Random changes to fs code to work around bugs in QDOS programs.

ERGON Development

Ergon Development have now released v2.24 of the Open World graphics conversion program as freeware. Along with Masterbasic 1.46 and the ZeXcel Spectrum emulator it is available for download from Ergon's Web site:

http://www.geocities.com/SiliconValley/Park/6533/davideeng.htm

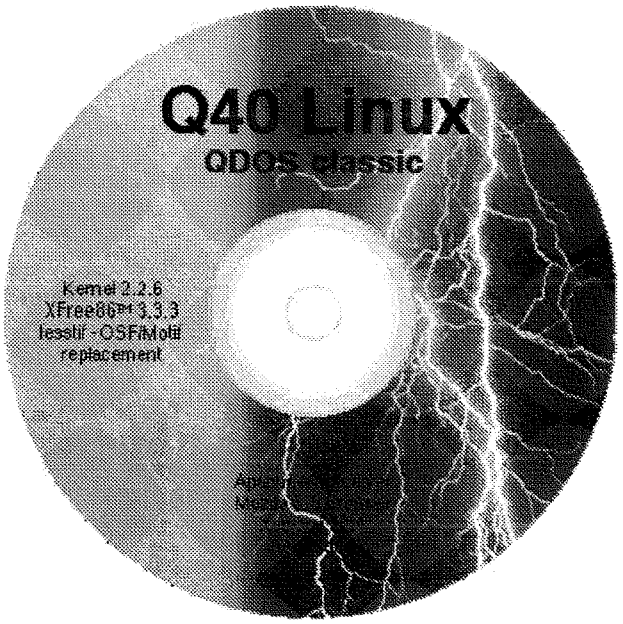
Davide Santachiara is looking at organising an Italian QL meeting later this year, possibly in September.

Q40 NEWS

pqiv011 is a picture viewer for the Q40 running under pointer environment. It requires SMSQ/E 2.95 on a Q40. PCX, PNG, PIC and GIF files may be viewed. qiv012 is also an image viewer, but for command line and which doesn't need pointer environment.

The Q40 web site **www.q40.de** includes the qs-player sound player software for the Q40 and Jonathan Hudson's sox sound conversion utili-

ties. Thierry Godefroy's port of POV3.1q (Persistence of Vision) ray-tracer software.
The Linux/QDOS classic CD-ROM is available via the online Q40 shop.



Jonathan Hudson News

Jonathan's Web site has moved - it is now on
http://www.hedwig.uklinux.net/index.html
Visitors to the previous address will be redirected after a short wait.
In addition to the "sox" sound file conversion utilities for QDOS systems, Jonathan has now released "Sound Toys", a sound playing utility which plays all sox supported sound formats (requires the sox package though).
Wxqt2 v0.06 is the 5/3/00 release of the graphical front end for qltools 2.14 and qxltool 1.13, which provides a graphical front end environment for accessing QDOS formatted media (floppy and hard disk) under Unix or Windows systems. qxltool1.13 addresses a problem copying files to/from the root directory on some Windows systems.
QLynx was updated on 3/3/00. This is the SMS/QDOS port of Lynx v2.8.2 broswer, which provides full internet/WWW access under the uQLx emulator. Lynx can convert HTML to ASCII. This version requires Gold Card or better - preferably uQLx.

TF Services

SMSQ/E 2.95e Beta and QDOS Classic beta version P are now available, along with the Linux/QDOS Classic CD-ROM for the Q40.
Work was progressing earlier this year towards release 1.99 of Minerva, which would add hires screen support under Aurora.

News from Dave Walker

Dave Walker's Web site now includes the 6/2/00 release of the C68 LIBC library in C68 for QDOS to fix a problem reported with scanf() routines. There is also a Beta version of Discover v4.20 that adds the ability to read CDROM drives attached to a Qubide interface, and a new release of the C68Tool utility for use with C68 for QDOS. Dave's site is on
http://www.itimpi.freeserve.co.uk/

QBranch

QBranch have loaned Tony Tebby an Aurora machine for colour driver development, stating that TT is already working on the development of SMSQ/E for QXL and QPC. QBranch and TF Services are looking into the feasibility of construction of the GoldFire expansion previously planned by Qubbesoft P/D.
QBranch have some of those rarities, ED disks, available for £15 for 10 disks.
We have a couple of QXLs, Qubides Super Gold Cards and Auroras for sale. Qubides, QXLs and Super Gold Cards are pretty rare at the moment.

ARCHIVERS CONTROL PANEL 4.00

is now available from Thierry Godefroy. New in this release:

- tar/gzip/bzip2/compress support added (these compression programs are also available from Thierry's Web site)
- New "tools" sub menu allowing splitting of large files into several fragments
- Extended configuration sub-menu

QemuLator News

Q-emuLator 2.0 for Windows 9x/NT is now available. It features a 640x480 full screen mode, very fast graphics. The QL display occupies most of the screen. You can also switch to 1024x768 full screen mode, where the QL display takes the whole screen. This is the slowest mode, due to the large amount of video memory used. It's the only mode that emulates flashing in the QL 8 colors mode. Faster accelerated text output and block graphics. It is possible to specify a key to be automatically pressed at the initial QL F1/F2 prompt.
Q-emuLator is shareware. This means that you can freely download a limited version of Q-emuLator and use it for two weeks to test many of its features and run your QL software (at limited speed). If you want to keep the emulator and unlock all the features, you buy a registration code that unlocks the program. Q-emulator can be downloaded from
http://www.geocities.com/SiliconValley/heights/1296/

Just Words! Web Page

Just Words! now has a web page at:
http://members.tripod.co.uk/geoffwicks/justwords.htm
This is a short and simple page to allow downloading of the demo versions of the program range.

RWAP News

A demo version of Flightdeck is now available, cost 2 pounds from myself. This has most of the features of the full version, except that flying time is limited to 10 minutes.
I have now released v3.5 of Payroll and v3.5 of Cash Trader which will mean that any small/medium size business can now once again use the QL to run off weekly/monthly payrolls and keep its accounts (including VAT reports). These programs cost 5 pounds each (either new or as upgrades from the original) or both can be ordered for 8 pounds.
I have also now re-released QL Cosmos, a former Talent astronomy program which allows you to display the position of the planets, stars and comets at any given time and date, from anywhere on the Earth. You can simply move a cursor over one of the objects in the sky to get more detailed information.
The SBASIC/SuperBASIC Reference Manual has now reached release 3. Each upgrade costs 6 pounds each for approx 40 double sided A4 sheets (or 10 pounds for 2 updates). Most of the Turbo Toolkit commands are now covered, including some which have never before been documented.
If you are uncertain which release of the manual you already have, please note that for Release 1, page 1 of the Contents is dated 10/1/98. Release 2 and 3 have the release number shown on the footer of the Foreword pages.
It is always worth updating the source disks - these cost 2 pounds.
I can also supply the updated version of the DIY Toolkit series (3 DD disks) for 4 pounds.

Q-Celt News

We have now released a CD-ROM for the QL of Line Design clipart. Over 5,500 _LDP files for Line Design, nearly 600MB of clipart in total. This CD should be suitable for QL emulators which can make use of QXL.WIN files (i.e. QXL, QPC, uQLx and recent versions of PC QemuLator). It is in ISO-9660 format. Regrettably the sheer quantity of clipart makes it impossible to make this available on other media. The price is £15 plus P&P.
Regrettably, the QL emulators CD-ROM compilation has been delayed, it is hope to be available soon. This will be freeware when it is eventually released, although please read the documentation as some authors may have placed minor restrictions on distribution of some programs included.
We have also released an updated version of the Dilwyn Jones Bargain Bundle CD-ROM, which includes a couple of new freeware programs and corrects some errors in the documentation. The CD costs £25 or existing users can upgrade for £5, it is of course still available on 11 HD or DD disks for the same price.
Q-Celt have also recently aquired a large quantity of high quality BLACK coloured PS/2 Keyboards, ideal for use on a PC (running QPC or whatever) to make it look more QL orientated!! They also work with the Di-Ren keyboard Interface on a real QL with a PS/2 to 5 pin DIN adaptor fitted to the end of the keyboard connector - they are untested as yet on SuperHermes. They are IBM branded keyboards, brand new and boxed, originally retailing at £27.99 each. We have good stock, but are selling fast. We are selling them at £15 plus P&P about half price.
We also have a quantity of second user lightly used ED disks for sale at £10 for a pack of 10. Anyone wanting larger quantities can contact Q-Celt to see if we can arrange a discount for larger quantities.

News from Bill Richardson

There are quite a few QLers who, with respect, we might refer to as the 'old brigade' looking for good old fashioned double density, or even high density twin drives as replacements. The problems are that DSDD are no longer made nor are Extra Density (ED). Sometimes HD drives can be used with DD disks, but some don't and are often incompatible in a twin drive configuration. as they are designed without a D1 D2 switch although this can sometimes be fixed, with a soldered link where there are pads to solder to on the circuit board. We have however just bought a quantity of TEAC high density drives which have not only got switches, but are compatible with double density and high density disks.
Regarding extra density,we have not been able to find them at all, and would be glad of any information on possible sources, and the same applies to discs although Roy Wood has found a few recently
The shortage problem also applies to RGB monitors, although we still have a few left, so dig out such products you have passed by and make the "old brigade" happy.

Cover Disk Notes

All files on this cover disk have been ZIPped because of the feature on Archiver's Control Panel. Have fun unzipping it all with Archiver's Control Panel - see Roy Wood's article. Once you have unzipped the various files, please read any accompanying _TXT or _DOC files. I suggest you LRUN the BOOT program first which will unzip one of the zip files of your choice (suggest you do Archivers Control Panel first). BOOT asks you to select one of the 14 zip files to unzip.

STEP 1

```
LRUN FLP1_BOOT
```

STEP 2

type in the number of the file shown on screen, from 1 to 14

STEP 3

enter the name of the drive containing the cover disk (e.g. FLP1_)

STEP 4

enter the name of the drive to unzip to (e.g. FLP2_)

If you have a single floppy disk drive system, you can unzip to ramdisk (RAM1_ or RAM2_) first, then copy all of the files from ramdisk onto a blank formatted floppy disk afterwards. I suggest you use a separate blank formatted floppy disk for the larger programs such as Archivers Control Panel and the RWAP games demos.

The files on the cover disk include:

ACP4e00_zip

Archiver's Control Panel v4.00 from Thierry Godefroy.

UNZIP

- you'll need this to decompress everything on this disk! This is the Jonathan Hudson Unzip v5.32 for the QL. The full QL versions of Zip and Unzip may be obtained from Jonathan's Web site

http://www.hedwig.uklinux.net/index.html

The BOOT program calls UNZIP to decompress the zip files on the disk.

GeeGraph_zip

contains the listings from Herb Schaaf's Gee Graphics series.

wolfboot_zip

contains the "My Boot" article from QLToday Vol 4 Issue 1

The following files contain the entries from the various authors for the CueDark modules competition. Some include text file notes.

Thierry_zip

contain modules from Thierry Godefroy for CueDark, but at his request these are simply supplied on an 'as is' basis, not actual entries for the competition.

Duncan_zip

contains two sets of modules, one set for the Q40, the other for the Aurora.

Stephen_zip

contains entries from Stephen Poole - most of these run best on a 512x256 QL screen.

Marcel_zip

contains a 'travelling through the stars' saver module, very effective on QPC as you'd expect from Marcel Kilgus.

Per_zip

contains a puzzle-type saver from Per Witte

Dietrich_zip

contains a floating digital clock routine from Dietrich Buder

INDEX4_zip

contains a text file index for Volume 4 of QL Today. Index of Volume 1, Volume 2 and Volume 3 along with back issues of previous cover disks can be downloaded from the Web site

http://www.soft.net.uk/dj/index.html

Gwasl13_zip

contains version 1.3 of George Gwilt's GWASL 68008 assembler, a more recent version of the program supplied on a previous cover disk. This version is better suited for use with Norman Dunbar's Assembler programming series.

QLTDIS3_zip

contains files for the QLToday Disassembler project as part of Norman Dunbar's assembler programming series.

MDP2_zip

contains Al Boehm's MidiPlayer 2 program, for those wanting to control a MIDI synthesiser from their QL network ports.

Flightdeck_zip

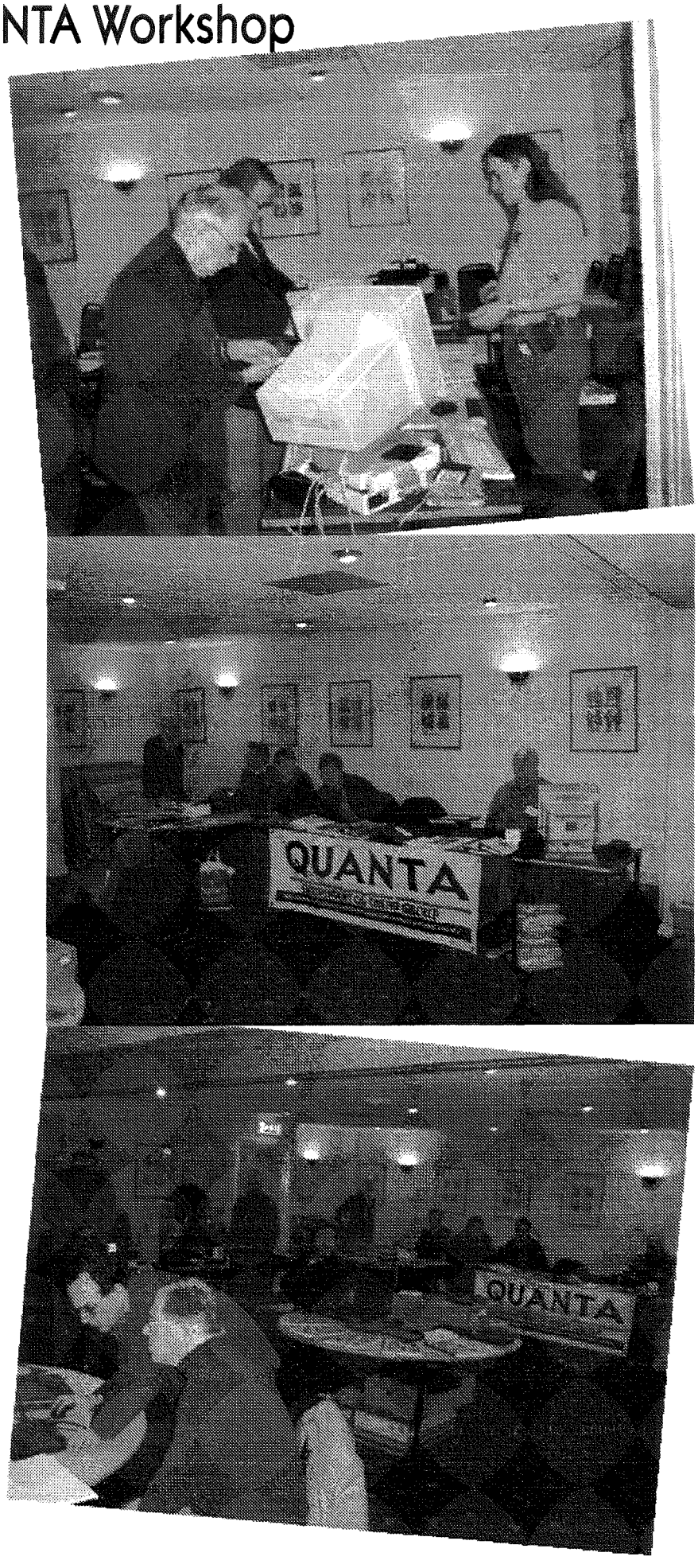
A demo version (time limited) of the Flightdeck flight simulation software from RWAP.

DdayMkII_zip

A demo version of D-Day MkII wargame from RWAP.

2000 Hove QUANTA Workshop

John Hall

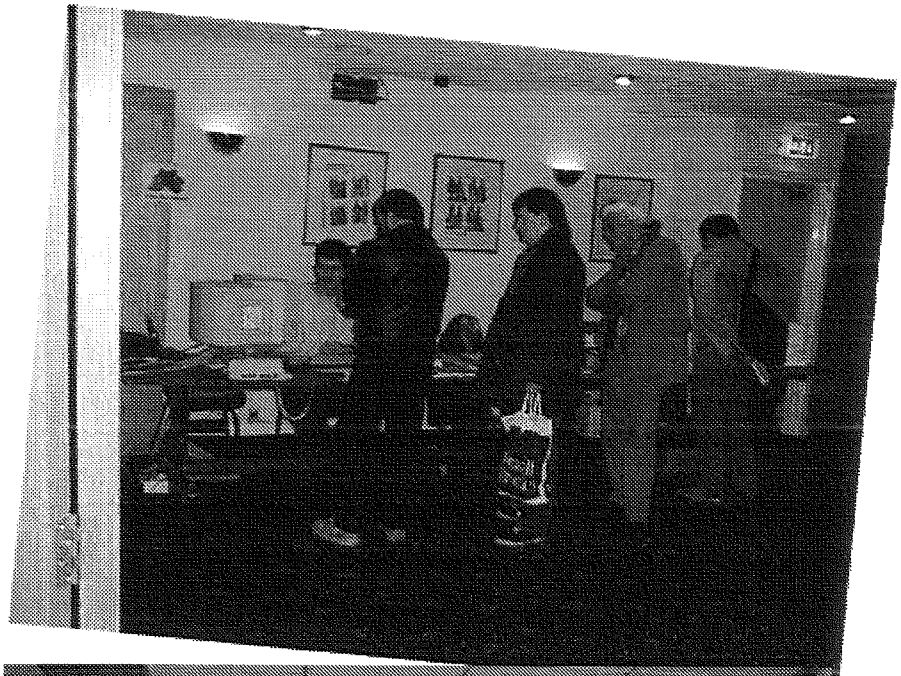


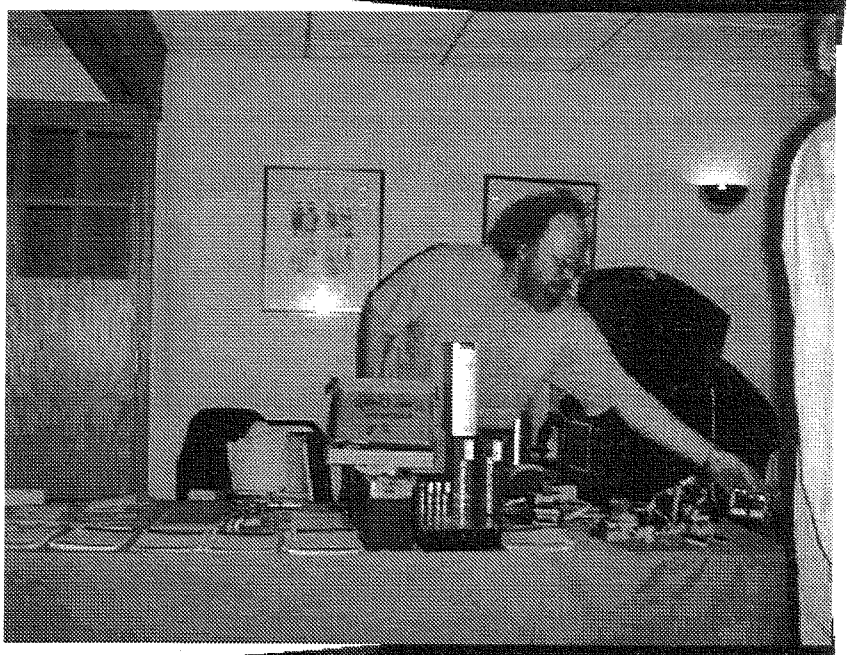
The 4th Hove QUANTA workshop was held on Sunday 27th February at the Excelsior Hotel on the Hove seafront.

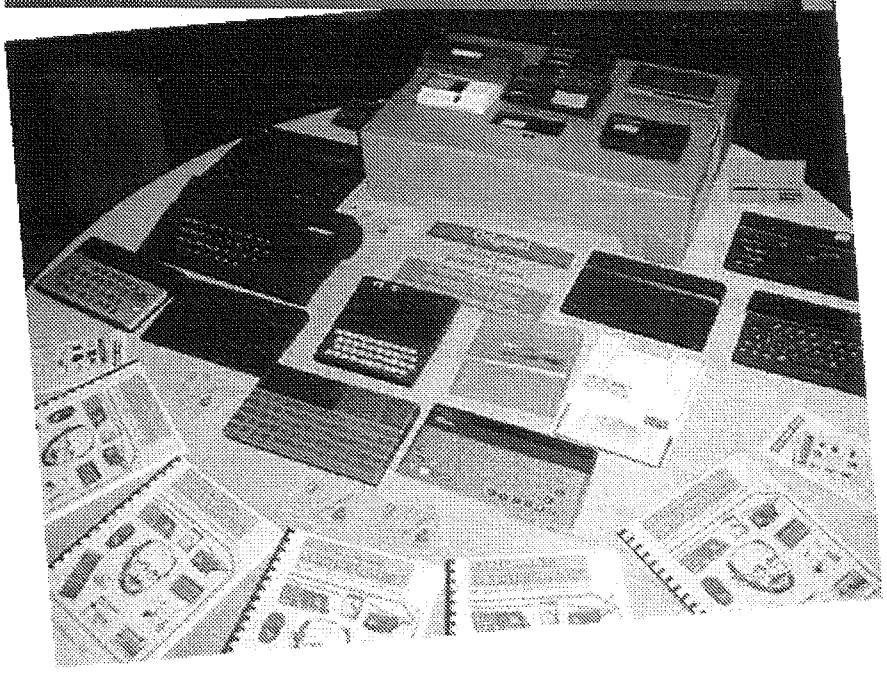
The gathering for the traditional pre-show Indian meal on the Saturday night was smaller than in previous years - just myself, Jochen Merz, Roy Wood, Roy's wife and their young daughter - and this was to an extent reflected in the turn-out at the show. For a mixture of personal and business reasons, Rich Mellor (RWAP Software), Darren Branagh (Q-Celt Computing) and Dilwyn Jones (QL Toady editor) were unable to attend. Ron Dunnett, who wound up Qubbesoft at the end of 1999, was also missed.

Those traders who were able to attend comprised Jochen Merz (JMS), Roy Wood (QBranch), Tony Firshman (TF Services), Geoff Wicks (Just Words!) and Bill (W.N.) Richardson. As last year, Enrico Tedeschi brought along his collection of Sinclair rarities and oddities and there was also a strong QUANTA presence, with Bill Newell, John Taylor, John Mason, John Gregory, Roy Brereton and Colin Baskett in attendance. Other QL notables who were around for all or part of the day were Jonathan Hudson (QTPI etc.), Chris Cave (MView), Mark Knight (Knight Safe etc.), David Gilham (PD Turbo Toolkit), Bruce Nicholls (ex-Quo Vadis) and Keith Mitchell (MinisQL).

Tony Firshman spent much of the morning building and tes-







ting a Q40 system for Jochen Merz. I understand that Jochen plans to transfer his Bulletin Board System, JMS-Box, onto it, which should be a good test of the reliability of both the hardware and the Q40-specific bits of SMSQ/E. As usual, Jochen was kept busy providing updates for his many programs, as well as promoting two new packages: Suqcess, a Pointer Environment front-end to DBAS, and Agenda, a Lotus Organiser look-alike that comes in both Pointer Environment and ProWesS flavours. Once again, Roy Wood had the unenviable task of both manning the QBranch stand, with it's comprehensive selection of QL-related hardware and software, and co-ordinating the running of the workshop.

In the afternoon, Roy Wood and Tony Firshman gave a talk on the current state of the Q40, which included encouraging news on the progress of the SMSQ/E colour drivers, plus a short demonstration of the Q40 Linux CDROM. Afterwards, Mark Knight gave a talk on his work on bug fixes and enhancements for The Editor, formerly published by Digital Precision and recently released into the Public Domain by its author, Chas Dillon.

The pictures on this page were taken from Enrico Terdeschi's website with his kind permission.

The QL Family Tree - The 90's

Dilwyn Jones

A look at the history of the QL from its launch in 1984 to the present day.

1990 Ablex finally discontinued production of microdrive cartridges, but later in the year restarted production when further supplies of tape became available. Miracle Systems announces a new 20% faster Trump Card using 1MB DRAM chips. The Thor XVI remained available for a time from Dansoft and Thor International, while David and Penny Oliver of CST went to America to work, having "run out of money" to continue in Denmark. Thor International closes down later in the year. QL World moved to the Maxwell stable in April. QJump launched QPAC2, Minerva got multiple BASICs and "a new concept" in QL software from Di-Ren, Fleet Tactical Command, allows multiple players to engage in a 3D naval warfare simulation on QLs over a network or modem. The same company launched a Micro Process Controller switchgear unit with switched relay outputs and driven from a parallel port. Di-Ren founder Robin Barker is currently better known as the chairman of Quanta. Supplier PDQL in trouble as Miracle Systems and EEC Ltd start court action over unpaid debts. PROGS launch Data Design, a unique pointer driven database system for the QL, later to become fully programmable. Meanwhile, yours truly sets up in business under the name of DJC to supply software for the QL, and the same year Chris Boutal launches QL Genealogist which was to become one of DJC's best selling programs as QL users take to researching family history in a big way.

1991 The first issue of the International QL Report is published by editor Bob Dyl sr. in America. Qubbesoft P/D's Ron Dunnett sets up a new PD library service. Digital Precision proudly launch their new Perfection wordprocessor (understating their software's names as usual!). A new QL group called NASA (!) is set up in Norway. The Amiga QDOS emulator is available. Level 2 drivers become available for the Super-QBoard and Trump Card, while Miracle Systems take the QL a giant step forward with the launch of the Gold Card, a 2MB 16MHz 68000 processor add-on card for the QL which boosts processing speed staggeringly. It also allows use of ED disks and drives for the first time. Though fast and with a capacity of 3.2MB (as opposed to 2.88MB on the PC) the disks are very expensive to buy and a little rare. Quanta takes over production of the QIMI interface and Cowo Electronic in Switzerland and QLympic systems in Germany announce a limited edition of 50 'SuperQL' machines based on Gold Card type QLs, later to be named the 'ExeQtor'. TF Services take over sales of Minerva from QView and advertise the Minerva MKII after a period when the original Minerva was mostly sold by word of mouth and bulletin board. January 1991 sees the first issue of the QL Hackers Journal - an e-zine for programmers by Tim Swenson, based in the USA. The C68 compiler becomes available, derived from PDQ-C advertised by PDQL but withdrawn by author Jeremy Allison, who removes the Lattice-C structures from it and the C68 we now associate with Dave Walker was born.

1992 QL World moved to Headway Home And Law Publishing briefly in mid 1992 then quickly on to a small publisher called Arcwind, whose most famous publication seemed to be a windsurfing and sailing magazine! Publisher Mark Kasprowicz had good intentions for QL World, which was still edited by Helen Armstrong as it had been for years. Grumbles appeared about TK Computerware at about this time and they eventually disappeared as far as the QL scene was concerned. Miracle Systems reduced the price of a Gold Card to £225 and sales went through the roof. Rumours abounded of a PC card from Miracle which would run QL software and of a new graphics card, and Digital Precision issued a new Gold Card version of Lightning SE which did away with the EPROM card because it could now run faster from RAM on the Gold Card. Ergon Development in Italy launched not one but two ZX Spectrum emulators running on the QL. Jochen Merz launched QSpread, the first serious alternative spreadsheet to Abacus for QLers, and also demonstrated QDOS running on an ST-QL emulator with a 68030 processor card running at 32MHz. The last major revision of Text 87, Text87Plus4, became available. A Quanta sub group (QLEA) released a clever little ROM switcher board which allowed your QL to switch between a Sinclair ROM and a Minerva giving users the best of both worlds - use a Minerva normally and switch back to a Sinclair ROM to run difficult old software or to test software on. Jürgen Falkenberg launches a new hard disk interface, the QL-HDD-Card. A new PD library service

HDD-Card. A new PD library service (Steve Johnson's SJPD) was launched and version 2.00 and 3.00 of the C68 compiler became available, both in the same year. TF Services launched the Hermes, a much improved replacement chip for the old Sinclair 8049 co-processor in the QL which became a must for many people. EEC Ltd (Bill Richardson's company as it was then called) briefly sold an infra red cordless serial mouse system for the QL after Albin Hessler released the first QL serial mouse driver software. Di-Ren released their QL-PC Fileserver software, indicating a trend of greater influence of PCs on the QL scene. Two ZX81 emulators, Xtricator and Xtender became available plus a Spectrum emulator called Spectator by Dutch software author Carlo Delhez. In early 1992 Tony Tebby had outlined a strategy with Miracle Systems and Jochen Merz to make a QL compatible version of SMS available, a concept which would borrow ideas from Tebby's Stella operating system in terms of modularity and so on - the project would come to fruition the following year. QLem, a little known QL emulator for some models of Atari ST and Falcon is released.

1993 Jürgen Falkenberg launches the QL2000 repackaged QL system based on a Gold Card QL and QL-HDD-Card system. Miracle Systems launch the QXL, a PC card using a 68EC040 processor chip with up to 8MB of RAM and a QL compatible network port. It is planned to use the SMSQ operating system from Tony Tebby and a derivative BASIC called SBASIC. Long delays and frustration are caused as it takes months for the full SMSQ to appear. Later, after some frustration on the part of Tony Tebby about what happened to the earlier SMS-2 which was never intended to be sold for the QL, he creates SMSQ/E and SBASIC (which unlike SMSQ for the QXL has built in Extended Environment and unlike SMS-2 has a BASIC interpreter) for all platforms including QL, QXL and Atari ST-QL. CGH Services' Richard Alexander finally quits as a QL trader in March 1993, having been a software publisher, PD library service and QL magazine publisher for many years. Ergon Development in Italy release the ZM/hT code compiling fast Spectrum emulator, which actually compiles Z80 code into 68000 code for faster execution, a real advance in emulation technology. Psion and Dansoft finally give permission for QL/Thor Xchange and the original QL Quill, Abacus, Archive and Easel to be essentially freeware for the QL scene, although this does not cover the PC versions. Bruce Nicholls launches QReview magazine for the QL through his Quo Vadis Design company, while Qubbesoft announce the Fastnet network system for the QL, reckoned to be 10 times faster than the original QL network. Hints are also dropped that Qubbesoft may produce an IDE hard disk interface for the QL. Obscure references to and rumours of a Tony Tebby operating system called Stella circulate, although Stella itself would not become available to QLers during the 20th century at least. Qubbesoft resurrect the original QJump QEP-III EPROM programmer, and this year sees the first 'Miracle In Newport', first of a series of annual shows in the USA.

1994 Qubbesoft P/D release the Qubide IDE hard disk system for the QL. DJC finally ships Page Designer 3 after one of the longest delays in QL software writing history! Miracle Systems launch the Super Gold Card early in the year with a 68020 processor, 4MB of RAM, a built in parallel port, a "crash proof" clock and more speed than had been seen on a QL this side of a QXL! After the delays of the QXL system software, the Super Gold Card comes as a breath of fresh air to QLers and proves to be the last of this type of expansion that side of the millennium. (The Ultra Gold Card and Goldfire were mooted later in the decade, but neither made it to market before the year 2000). Di-Ren launch the ill-fated System Amadeus, a token ring expansion system designed to allow peripherals to be daisy chained from the EPROM slot of a QL. The system promised parallel ports, fast serial ports, DIY interfaces, sound sampler and playback, but was eventually abandoned later in the year. QL World finally closes with the May 1994 issue and IQLR (International QL Report) moves to fill the gap left by QL World's demise. The very last article in QL World was Linear Regression by A.F Wilson, with an advert from yours truly on the back page, so it could be said I was the one who had to close the door and switch the light out on QL World. Not sure I want to be remembered for that though. Jonathan Hudson releases QTPI, a pointer driven communications program which was to have a huge impact in following years. SMSQ/E was released for QL and Atari, along with a version for the QXL. DV3 (the third standard of disk access, DV2 gave us 'hard directories', gave us access to PC formatted floppy disks from SMSQ/E). The Level 2 Config specification is released. Much prompting persuades Norman Dunbar to write 'The PE Idiots Guide' which by itself resulted in a mass move of QLers into using QPAC2 and the pointer environment.

1995 DJC ceases trading, but out of the ashes spring three new software companies - QBranch (Roy Wood), Quo Vadis Design (Bruce Nicholls) and Geoff Wicks, later to adopt the name 'Just Words'. QReview magazine merges with IQLR, leaving Bruce Nicholls free to concentrate on software publishing. Eros Forenzi introduces the QSI standard speed tests for QL based systems. Digital Precision makes all its software available in one giant bundle for only £149 with on-disk manuals in a final push for sales before stopping advertising their QL software the following year. Miracle Systems start to sell software - for a while. Di-Ren release a low cost and simple to install keyboard interface using programmable logic chips. TF Services release the SuperHermes board, an enhanced Hermes with fast serial port, mouse and keyboard interfaces. Qubbesoft P/D release the QPlane, a buffered backplane to fill the needs of the increasing number of people transplanting their QL systems into PC-style cases. Daniele Terdina releases the first QL emulator running on an Apple Mac, called simply QemuLator, and proposes a PowerMac version. Simon N. Goodwin writes that he has become the first person to run QDOS on a 50MHz 68060 Amiga accelerator, using Amiga QDOS v3.23. Simon also produces the first QDOS audio CD player on Amiga QDOS. PROGS announce that they are working on ProWesS - a completely new window manager system for the QL. Dennis Donahue documents for the first time how he made a CD-ROM of all the QL software he could find simply by creating a large QXL.WIN on his hard disk and copying it onto CDR using a standard PC CD-Writer, but finds that QL software is too compact and he is unable to fill all 600MB+ he could theoretically get on a CD for use with his QXL! It would be 1999 before the first commercial QL software became available on CD (unless you count the Amiga Format cover CD in 1996 where Simon Goodwin compiled tens of megabytes of QL software for the Amiga QDOS emulator in 1996). Qubbesoft Miracle Systems announced a new product called QXL-Gold which would allow a Super Gold Card to be plugged into a PC, but sadly the product does not become available in the end. The QXL is discontinued because of rising component prices and not because of the incorrectly rumoured worries about the European EMC directives. Nasta demonstrates a mono LCD screen running with a QL. Michael Klein proposes the development of Primus, a 19 inch rack mounted QL derivative for industrial environments. People start overclocking QXLs to make them faster. September sees a 486 software based QL emulator demonstrated at a German meeting by a young programmer, who is immediately signed up by Jochen Merz to produce what would later be called QPC.

1996 Geoff Wicks launches the first ever QL Thesaurus and Style Checker programs. Ill health eventually forces Bob Dyl to cease publication of International QL Report, and Jochen Merz and Stuart Honeyball decide to start a new magazine called QL Today and a desperate search for an editor fails and they are forced to appoint this mad Welshman as editor of the English version (there is a separate German edition). After more than a year's work, PROGS of Belgium finally launch ProWesS, the new Window Manager system for the QL. Qubbesoft release the Aurora replacement motherboard incorporating enhanced graphics capabilities, though it was to be some time before SMSQ/E supported the enhanced display resolutions and indeed the next century before the higher colour modes were supported. Qubbesoft are supplying 135MB collections of software on EZ135 cartridges for Qubide users. QPC1 is launched. Quanta sells out of the first batch of Super Gold Cards made after Miracle discontinued the product. Miracle relaunches the QXL as the QXL2 with a 25MHz 68EC040 processor and Terry Harman goes one better and recounts his experience of overclocking one to 80MHz! Simon Goodwin permits Qubbesoft P/D to supply the entire set of disks of the long running QL World series DIY Toolkit to be supplied at PD library rates for the first time. TF Services and Bill Richardson decide to make available the serial to parallel printer converter interfaces once more. Wolfgang Lenerz releases an SBASIC interface for ProWesS. SJPD released the first QL encyclopedia on disk (or 3 disks rather). Based on the Probert Encyclopedia, it was freeware! Richard Zidlicky was working on a QL emulator for Unix/Linux systems, called uQLx, after testing 68000 emulation on the QLem emulator running on STonX on a Sun Sparc system and proving himself wrong that a successful 68000 emulation in software would not be possible. Richard bases his work on Daniele Terdina's QemuLator source code. SERNET is released, developed from Midinet, this allows a QL network style connection via serial ports between PCs, Ataris, QLs etc. 4 years later people were still struggling to get the cabling right!

1997 Miracle Systems enter into an agreement with QBranch, allowing QBranch to handle all sales of Miracle products, including the remains of the second batch of Super Gold Cards made by Quanta. SJPD announce their closure in January 1997, then stages a welcome return a few months later. QL Today issues its first cover disk (I think the honour of the first QL magazine to have a cover mounted disk went to QReview in 1994 or 1995). Simon Goodwin (where would the QL scene have been without him over the years?) wrote that it was now possible for QDOS/SMSQ software to take advantage of hardware floating point units using some code written by George Gwilt - although the first QL type hardware to use FPUs was the somewhat rare "Thor 21" years previously. QLay emulator (its 68000 emulation code is derived from UAE, an Amiga emulator by Bernd Schmidt and others) is launched, along with a PC version of QemuLator towards the end of the year. A lot of QL hardware is announced during this year (Ethernet Card, Sound Card, Goldfire, bidirectional parallel ports, Ultra Gold Card) which never actually makes it to the marketplace! Quo Vadis Design closes as a software retailer, but Bruce Nicholls plans to continue to be active in the QL scene - he goes on to set up the now popular QL Users Email Mailing List. The Manchester QL user group sell Pandora - a custom built black steel case to house QL systems. TF Services release SuperHermes Lite version of their existing SuperHermes. Nasta generates a lot of interest with in depth articles on multi-processing. TF Services announce the RomDisq, designed by Miracle Systems, a tiny 2MB or 8MB flash memory card for the QL which simply plugs into the EPROM slot of a QL. Rich Mellor announces that QBranch will sell his opus magnus, the first QL route finder software, an application which users had been almost begging to get on the QL for some time. QLay becomes a Windows 95 version, probably the first QL emulator on a PC to run in its own window, not a DOS box.

1998 Rich Mellor sets up in business as RWAP Software. Digital Precision allow work to begin on updating some of their old software such as the Turbo Toolkit, work which keeps Mark Knight, David Gilham and others out of mischief for a couple of years. People worried about the impact of the 'millennium bug' on QL systems are surprised to discover that in fact our equivalent will in fact happen in 2029 or 2097 depending on whether our systems use signed time counters or not. QBranch and TF Services join forces to launch the MinisQL, a cased Aurora system using laptop sized Epson computer cases, allowing integration of a Gold Card, Hard disk and so on. Richard Zidlicky's uQLx emulator handles the highest resolution screens yet for a QL type system - 8192x4096 - although I'm not sure where he found a monitor to display this resolution! Bruce Nicholls sets up the QL Users Email Mailing list, reflecting the growing number of QLers who use PCs or other computers in addition to QL and derivative machines. Although the QL has had a presence on newsgroups for some time, this is the first such forum organised via simple email, making it easier for many people with some form of access to email only to join in. Francois Lanciault announces Paragraph, the first pointer driven wordprocessor for the QL, based on the ProWesS engine. Jochen Merz announces the Milan computer as the next QL compatible system but no version of SMSQ/E becomes available for it and the machine seems to quietly disappear from the QL scene. QL scene has to face up to how to handle the Euro currency symbol. A very rare entity, a badly written QL virus program makes a brief appearance at a Quanta show that year, proving that we are not 100% immune from this curse. Rich Mellor completes the SBASIC/SuperBASIC Reference Manual - a massive 1,000 page reference guide to QL BASIC. Quanta take great pride in reporting that a large forecast deficit in their accounts had in fact been overturned thanks to drastic cost cutting measures by the committee. Work by Jonathan Hudson and Richard Zidlicky mean that the Lynx browser running on uQLx on a Unix system can now access the underlying Unix IP sub-system to access the internet. In plain English, that means uQLx became the first QDOS platform where you could access email and the Web. QLCF set the ball rolling on colour drivers by contributing hard cash to Tony Tebby. SJPD finally closes 1st August this year. Amstrad issue permission to distribute QL ROM images in Europe. The Q40 system is finally pushed towards market, and Mark Swift works on porting Amiga QDOS to a form known as QDOS Classic Q40 as an OS for the Q40. TF Services launch the MPlane, a low profile backplane card designed especially for the MinisQL. Qubbesoft launch the version 2.00 Qubide ROM, which allows Qubide users to access Iomega ZIP drives.

1999 SJPD found a new owner, in Phil Jordan, who sets up The Library PD Service and Darren Branagh sets up Q-Celt Computing, proving it's still possible to set up in business as a QL trader. Sadly, Qubbesoft P/D ceases trading in December 1999. Jochen Merz releases QPC2, an enhancement of the original QPC long requested by users. Chas Dillon releases sources for his former PDQL and Digital Precision programs, including Turbo, The Editor, and some business programs, so some eminent QL programmers set to work to produce updated versions of these classic programs. Q40 now has a choice of three operating systems - QDOS Classic, SMSQ/E and Linux (the latter thanks to Richard Zidlicky). Q-Celt Computing becomes the first company to issue commercial QL software on CD-ROMs for those emulator users who can access QXL.WIN structured media. The QL scene nearly loses a number of its traders in an accident prone trip from England to a show in Croatia - despite a potentially serious car accident, they return to tell the tale with only a damaged Volvo as evidence. Work starts in Switzerland on a TCP/IP system for the QL. Sadness as one of the best known and longest established Quanta subgroups (Bristol) calls it a day - this group was responsible for the Bristol area Quanta workshops, among the most successful QL shows in recent years, though shows will continue to be held in the area from time to time. Giorgio Garabello sets up the Quantum Ring, an interlinked set of QL Web sites as QL presence on the Web takes off in a big way. Colour Drivers, or GD2 or Graphics Device Interface 2 to use an 'official' term for them, become available for the Q40 first. Club QL International quietly shuts down, a victim of falling membership due to the growth in online communication. Simon Goodwin and Al Boehm bring out a QL Midi player done exclusively in software via QL network ports. Long time North American QL dealer Frank Davis hands over the reins to John R. Rish of Home Electronics Service in Texas, who becomes the QL's man in America for the next century! The first QL JPG viewer is written by Dave Westbury - a program sorely needed now that the colour drivers are here. Great interest is shown in Tony Tebby's Stella system thanks to a virtually single handed campaign by French QL user Arnould Nazarian. And just when we thought we were having a good year, the Quanta committee embarks on a proposal that Quanta should open its doors to other computer users and become a multi platform user group. Views are mixed, though most agree that action is required to stem falling membership even if this is a bit too drastic a way of going about it. Jochen Merz becomes the first QL trader to set up a Secure Web Transactions service, allowing QL goods to be bought over the Web.

Of Webs Echoes Leaps and Surfing Black Boxes

J. Dent

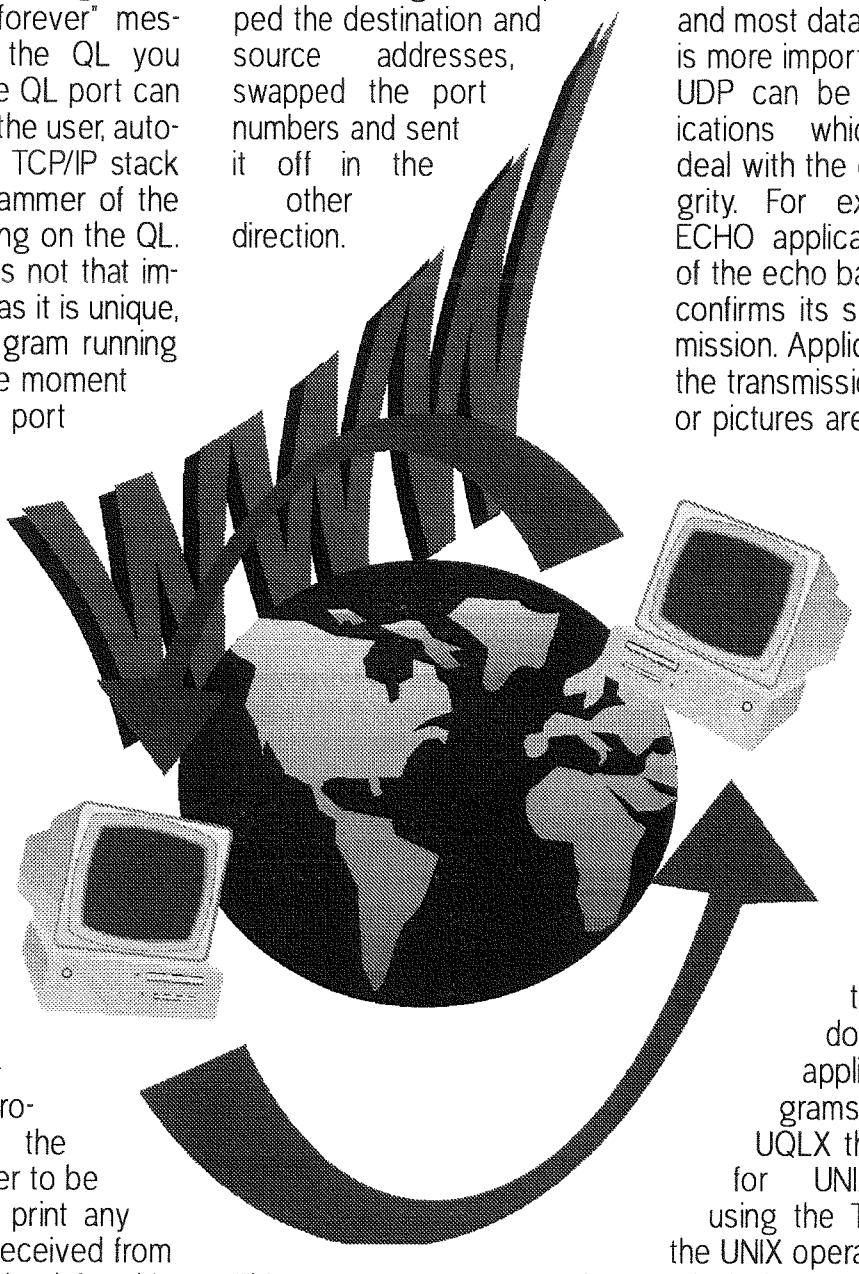
During November of 1999 unnoticed by the rest of the world a datagram echoed through the Internet carrying the message "QL forever". A small step for a datagram but a quantum leap for mankind. Well perhaps not quite for the whole of mankind but at least for the kind of man (or woman for that matter) who prefers his little black box to one of those Pentathlon things or whatever they're called these days.

What I'm referring to is the first test transmissions using the TCP/IP stack I'm developing for the QL and its derivatives. We are told that the 21st Century will see practically everything connected to the net: telephone, TV, central heating, cooker, fridge, washing machine or whatever turns you on, or perhaps more to the point whatever you may wish to turn on, off, up, down or adjust, have serviced etc... It would have been a shame if the list did not include the QL.

The QL isn't going to be left out. The tests prove that communication over the Internet is possible. The protocol, which was used for the transmission, is known as UDP. That's short for user datagram protocol. The datagram is the whole packet of data, which is transmitted through the Internet. It includes the original user data in this case "QL forever" and IP and UDP header information. The IP header information includes the Internet address of the destination computer and of the source computer, in this case the QL. The address of the source has to be known by the user. The address of the QL is allocated by the ISP (Internet Service Provider) and may well be different every time you dial up. The IP header

also indicates the length of the data being transported. The UDP header contains the number of the port on the remote computer and the number of the port on the QL. The number of the port on the remote computer depends on what application you want to use on that computer. For instance to get it to echo the "QL forever" message back to the QL you need port 7. The QL port can be allocated by the user, automatically by the TCP/IP stack or by the programmer of the application running on the QL. Its actual value is not that important so long as it is unique, i.e. no other program running on the QL at the moment is using this port number.

For the above test the application is a simple program which allows the user to enter the address of the remote computer, the port of the target application and the port number on which the QL will be listening. A separate, receiver, program requires the same port number to be entered and will print any UDP messages received from the Internet destined for this port. A message, like "QL forever" (what else) can now be entered in the first QL program and lo and behold it appears in the screen of the receiver program. It has been sent to the local ISP, his computer has checked the address and sent it onto the appropriate part of the Internet and maybe handed from one network to another until it arrives at the destination computer. The ECHO application on this computer has received the datagram copied the "QL forever" message to a new datagram, swapped the destination and source addresses, swapped the port numbers and sent it off in the other direction.



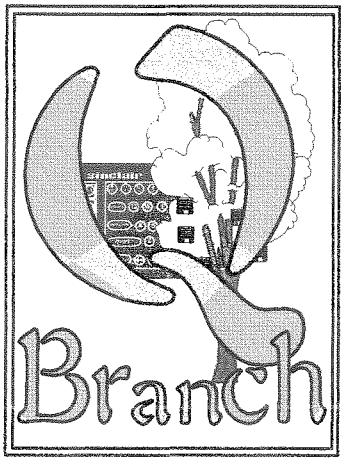
This is the simplest type of protocol for use on the Internet, it is now running on the QL and it is an important protocol but the most important protocol is TCP. TCP stands for Transmission Control Protocol. Whereas when you send a UDP datagram onto the Internet you have no idea if it arrives at the other side, if you send it by TCP then TCP will ensure that it arrives repeating it if necessary. If all TCP attempts to transfer the data fail then you are warned.

TCP is used for transferring files, World Wide Web pages and most data where integrity is more important than speed. UDP can be used for applications which themselves deal with the question of integrity. For example in the ECHO application the arrival of the echo back at the origin confirms its successful transmission. Applications requiring the transmission of live sound or pictures are well served by UDP's simplicity and speed. They gain no advantage from the reliability of TCP. A repeated block of such data will arrive too late and is of no use.

I'm working on TCP now. About 60% of the coding is done. There are application programs running on UQLX the QL emulator for UNIX computers using the TCP/IP stack of the UNIX operating system on which the emulator is running. These include Lynx a browser for the World Wide Web and e-mail handling programs.

So get ready to leap into the 21st century and take your black box surfing.

Q Branch



PROGRAMMING

QD 98	£ 53.00
QD + QBasic	£ 69.00
QD + Qliberator + QBasic	£ 110.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 37.00
Easyptr pt 1 & 2 (together)	£ 33.00
Easyptr pt 3 (C library)	£ 16.00
QMake	£ 18.00
QMon / JMon	£ 22.00
Basic Linker	£ 22.00
DISA 3	£ 37.00
QMenu	£ 16.00

Text 87

£ 79.00

Typset 94 £ 29.00
Fountext 94 £ 39.00
2488 drivers £ 29.00
Epson ESC/P2 drivers £ 26.00

Text 87 is the only QDOS / SMSQ wordprocessor capable of handliing the full screen on the Aurora / QXL / QPC systems. New drivers are currently being written.

Shop hours :
Mon - Fri
10am - 6pm
Saturday
10am - 1pm
Wed - closed

UTILITIES

FiFi 2	£ 22.00
QSup	£ 32.00
QSpread v2.04	£ 66.00
Cueshell 2	£ 30.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 20.00
QPAC 2	£ 40.00
QTYP 2	£ 30.00
QLQ	£ 32.00

'Just Words' by Geoff Wicks

THESAURUS, STYLE CHECK, SOLVITPLUS 3

£ 10.00 ea / ANY 2 PROGRAMS £ 18.00 / ALL 3 PROGRAMS £ 25.00
(Includes Pointer and non-pointer driven versions)
(P.E. versions need Hot_rext, WMAN and PTR_GEN or SMSQ/E to run)
Upgrades from previous versions £ 2.50 + S.A.E. New Manuals £ 1.50

QL2PC - New program !!

Convert text files from QL to PC formats and much more !

Only £ 10.00 Now with HTML support !

Spelling Crib : PD program £ 1.50 +SAE
or Free if you buy all three programs

The colour drivers for the Q 40 are now at version 2.97 and the support disk for this has been changed a lot since it s issue. Please return your disks for upgrading (include return postage).

While you are at it why not send off for a copy of ProWesS now that the base installation is free ! Look at the prices on the other page. You can now get LINEdesign with the free ProWesS installation disk for only £24.00 and you can give Paragraph a whirl too !

It seems that Tony Firshman and I will be taking over the GoldFire project now that Qubbesoft have dropped out. We may also be taking over the Aurora and Qubide as well - more details in the next issue.

Now is a good time to upgrade your system to an Aurora because Tony Tebby is working on the drivers for that system. We still have a couple of these in stock if you want one and we may be building some more soon.

Another new program is added to our list with SuQcess a great new database program. This will import Archive files and is very user friendly. Come and see it and the Q 40 at the Quanta AGM in Manchester in April

We are currently out of stock of the SuperBasic Reference Manual Place your order now to get one as soon as it is reprinted

The SBASIC / SuperBASIC Reference Manual

The complete definitive guide to BASIC programming in QDOS / SMSQ including three disks of PD toolkits, example procedures and an electronic index.
compiled by Rich Mellor, Franz Hermann and Peter Jaeger

Over 500 pages !
£ 40.00
+ postage

Q Branch

Feeling out on a limb ?
Reach out for Q Branch.
Suppliers of Quality QDOS/SMSQ products
Hardware and Software.

Tel +44 (0) 1903-200212 fax +44 (0) 1903-208070
email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk

The Bank Volt, 6 Coronation Buildings
Ham Road, Worthing.
W. Sussex. BN11 2NN. UK.

Q Branch

Hardware

We have a small stock of second user items. Auroras / Qubides / Gold Cards / Qplanes / superHermes etc. call us to get details of the items available. These are going fast so call soon.

only Few Super Gold Cards	£ 125.00
QXL	£ Call
Recycled superHermes	£ 80.00 *
Recycled Gold Card	£ 60.00 *
Recycled Aurora	£ 75.00 *
Aurora	£ 90.00
Qubide	£ 55.00
Qplane	£ 25.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
'Arfa Braquet'	£ 8.00
'Son of Braquet'	£ 18.00
The 'Braquet'	£ 16.00
MC plate	£ 6.50

* when available.
14" and 15" monitors for the Aurora - Call.

ProWesS

ProWesS (now free !)	£ 1.60
DATAdesign	£ 24.00
Fontutils	£ 30.00
File Search	£ 12.00
PFlist	£ 12.00
Dilwyn's Fontpack	£ Call
LINEdesign v 2.16	£ 24.00
PWfile	£ 18.00

Paragraph

The ProWesS word processor

Demo version £ 1.50 + postage
Full Registered version £ 18.00

Version 2.02 available now !

Q Branch Programs

The Knight Safe 3 *NEW VERSION!* £ 35.00
upgrades from previous versions £ 5.00

Q - Route v1.08C £ 25.00
Route finding programme

Q - Count *NEW VERSION!* £ 25.00
Pointer driven home accounting

The Fractal Collection !

This is a brand new program which will produce stunning animated fractal patterns. It will run on anything from a Gold Card to the Q40 and will be capable of using the power of the colour drivers when they are released. Complete with many example files and routines to design your own screens.

Only £ 35.00

SMSQ/E

Gold Card / Atari / QXL Version

£ 76.00

Various Atari versions : call for details

The Colour Drivers are running on the Q Branch Q 40 !
Coming to a computer near you in 2000!

QPC 2 is here !

£ 90.00 (£ 70.00 SMSQ/E Owners)

Upgrades from QPC 1 £ 30.00
(return master disk)

Special offer !
Get Cueshell for only £15.00 with any copy of SMSQ/E

There are so many extras available for the Q 40 that it is hard to list them here. If you are interested in one of these boards than contact us and we will supply the details. the basic prices are shown here.

Q 40 board with 16Mb RAM & I/O card	£ 330.00
+ SMSQ/E	£ 30.00
Extra 16 Mb RAM	£ 30.00
Tower Case	£ 45.00

All prices do not include shipping.

We can accept payment by VISA, Mastercard and Switch. You can also pay by Eurocheques made out in Sterling or a Sterling cheque drawn on a UK Bank. Prices include Post and Packing in Europe.

Q 40 Tips

Roy Wood

The recent Eindhoven show provided a few hints and pointers for Q40 users. One of these concerned the use of MKPART_EXE the program that is used to create partitions on hard disks. First you should use v 1.02 of MKPART_EXE or later if it is available. Earlier versions have bugs which could cause problems later on. I have included this version with the current distribution disk and you should return your disk for a free upgrade if you do not have it. I have also expanded the notes for using it to make it easier.

One thing the documentation did not make clear is that the size of the QDOS partition is critical to the performance of the machine. If you have a large hard drive, I had a 1Gb when I first started out, and use it as one partition the slave block handling starts to slow everything down as you put more onto the disk. It is better to use a maximum size of 256Mb to keep the machine performing well. The drives can be linked and unlinked from the system in the same way as the Qubide drives. If you use the WIN_DRIVE command you can set which partition is linked in with which drive indentification. The first partition is always linked in as win1_ as a default when SMSQ/E is started but from there it is up to the user. The format is

```
WIN_DRIVE WINn_,DRIVEn,PARTITIONn
```

for example:

```
WIN_DRIVE 2,0,1
```

Create WIN2_ using the first Hard Drive and the second partition.

```
WIN_DRIVE 3,1,0
```

Create WIN3_ using the second Hard Drive and the first partition.

Tony Tebby tells me that the 'The BASIC Atari partition table (requested by Peter) has 4 entries. There are at least two extended partition tables but I do not know if any other system for the Q40 would support them.'

This means that you can only create 4 partitions on a hard disk for use by SMSQ/E. If you use MKPART.EXE to create partitions of 256Mb as suggested on a drive larger than 1Gb the last partition will be forced into being of a higher value than the optimum suggested.

This is because you can set up a second hard drive as a slave. There seems to be no mechanism to partition a second hard drive from MKPART_EXE so it is better to set it up as a master and partition it before you link it in as a slave drive.

One other thing that has been causing a bit of confusion now that the Colour Drivers are finally reaching a wider public is the way they use the resolutions. I cannot recall seeing this documented anywhere but it would seem that the lower resolution of 512 x 256 pixels does not use the colour displays at all. If you move up to a higher resolution then the colour comands all seem to work you can use the commands.

I also learned, at the Hove show, that ProWesS is very suspectible to which mode it starts in. If you load ProWesS in the 512 x 256 mode at the start of your boot file and then move the display to 1024 x 512 the mode 33 driver is not linked in and the display is scattered across the screen. It is far better to start the system is a higher resolution if you want to use ProWesS. This is where the branching boot file comes in handy. You could offer an option of which display or set of programs you need and then select certain procedures within the BOOT. Alternatively make the ProWesS extensions load in a separate BASIC program and call that when the system is already in Hi-RES mode.

Image Viewer PQIV

Claus Graf

There is a picture viewer program for the Q40: pqiv. It uses the Q40's display modes with 65535 colours, therefore it demands at least SMSQ/E 2.95. At the moment it's possible to read and display the following image files:

- PCX
- PNG
- PIC
- GIF

The program is written for the Pointer Environment and can hopefully be used without a manual. Next step to do is the possibility to save pictures. That means that PCX, PNG or GIF files can be converted to PIC or saved as a background image. Then you can load nice backgrounds on your Q40 with the SBasic command BGIMAGE. PQIV in the current version is free and can be downloaded from the Q40 Website **http://www.q40.de**.

Jpegs how to get some - or a review of a Kodak 215 zoom digital camera

Bill Waugh

Dave Westbury is working on software that should allow us to view Jpeg graphic files on Qdos and sms systems, I have an early version of Dave's software but as yet have not had time to try it, due in part to a recent 16 day holiday to the west coast of USA and a multitude of other commitments, isn't it amazing how stuff comes along to fill any spare time that you thought you were going to have?

I have been thinking of buying a digital camera for some time now but have been put off by the price of the ones with the best resolution or the poor resolution of those that I found affordable, however the above mentioned holiday and the chance to purchase a Kodak 215 zoom at a very good price gave me the incentive to bend some plastic, I don't want to go into detail as to where I purchased the camera but the words holiday and Walmart might give you a clue, suffice to say that the price of the camera and an additional 16 meg storage media was about £100 cheaper than you would pay at Dixons in Britain.

Before I describe the functions of the camera let me say this, I was a bit concerned about the resolution. The most recent cameras on the market have resolutions in the order of 3 megapixels and can cost up to a thousand pounds, the Kodak 215 has 1 megapixels and can be bought at Dixons for £260 - of course you may get it cheaper or haggle a bit.

Is it good enough

Kodak state that it will give photo quality on a printout of 7x5 inches, I can say that this is accurate, I have printed some of the photos I took on an Epson 740 (fit to page) and they are excellent at 8x6, and the colour is amazing, so if you have been waiting for digital cameras to come of age then I would say you could buy now.

If anyone would like to see an example of a photo print out then I could send one out to you, bear in mind though that glossy paper is still about 50p a sheet and colour cartridges cost also, so a pound coin included with your name and address would be appreciated.

The camera saves the images in Jpeg format or FlashPix, the supplied media is 4 megabytes and will hold 12 images in highest resolution 1152x864, and best Quality.

Colour is 24 bit (millions of colours).

Resolution can be set at 1152x864 or 640x480.

Best quality needs explanation, it really means the degree of compression, the more compression used some detail or colour will be lost so less compression = less pictures stored but better detail kept in the pictures.

I purchased a larger 16 megabyte media, it holds 62 images at high res best quality and came with a USB reader. USB is a newish sort of serial port for PC and allows quicker down load of the images.

The camera is quite neat in size, about what you would expect for a normal camera. The front has the telephoto lens (2x zoom), flash, viewfinder, light sensor, flash sensor and an indicator for the self timer. On top of the camera are four buttons Shutter, self timer, close up (macro), and flash, also there is the status display, this indicates quality and resolution choice, battery state, flash type chosen, red-eye, close-up, camera memory card and self-timer.

Viewed from the back the right hand side has a slide to release the battery cover, while the left side has a door for the memory card and three rubber flap covered jack plug ports, the top one is the serial output, middle one is for an ac adapter and the third is video out.

The back of the camera has as you would expect a Lcd display, a power switch and next to it a status light, a slider to select zoom, four scroll buttons, a " do it " button and a slider with the following positions

1. Capture - for taking pictures
2. Review - for viewing pictures you have already taken
3. connect for downloading pictures via a serial cable or video cable to view your pictures on a TV
4. Preference - this is where you set the camera to your individual requirements.

Taking pictures

This is very much point and click, the camera does it all for you although there is the facility to set exposure to constant should you wish to take a series of images with the same exposure setting. Flash - can be altered to one of five types

1. Auto - default
2. Fill - always on
3. Red-Eye - eliminates the same

4. Fill + Red-Eye
5. Off Exposure - as mentioned the exposure can be locked to give a consistency it can also be altered from -2 to +2 in .5 increments an increase of 1 doubles the amount of light allowed into the camera.
Preview - you can select to preview your image before you take it, this does consume batteries, also available is "Quick preview", the image is displayed for a few seconds after being taken this allows you to delete and retake the image again should the inevitable thumb before lens syndrome strike.
Templates - there are templates on the supplied media and you can choose to add a template to pictures taken, these however can not be removed once selected, there are no templates on the third party media I bought.
Date - you can date stamp your pictures.
Image type - select between FlashPix or Jpeg

Displaying pictures

Computer - software is supplied for Mac or PC to display, enhance, store and print images, as the camera has a serial link hopefully we might get someone to show us how to download them onto our Qdos/Smsq based systems.
TV - a cable is supplied that connects to a video port on a TV (jack plug type), after selecting an AV channel pictures can be displayed one at a time or as a slide show with each picture displayed for 5 seconds. TV output is selectable between PAL and NTSC.
You may have gathered from this review that I am pleased with this camera, while it may not have a resolution that allows for A4 printout this is not something I will miss very often and I am more than happy with the suggested print size of 7x5.

It would also have been nice to have been able to fit a different lens at sometime but the range of the supplied zoom lens is really quite adequate.
I have no interests with Kodak Eastman and have written this review with the sole aim of informing QL Today readers that if like me you have held off from buying a digital camera because you are worried that they are still developing and your purchase might prove disappointing in a years time then take a look at them now, the mid range priced ones might just be what you require.

[I can fully agree - I bought a second-hand Olympos with similar specs, 1.3 million pixels, and the results are pretty good - Jochen]

Archivers Control Panel

Roy Wood

In the last issue I wrote a short guide to FileInfo II. I feel I should follow this by giving you another guide to the Archivers Control Panel. This program is a boon to those of you who, like me, can never remember strings of command letters to get zip and unzip to work. This is compounded by the fact that there are several other compression programs available to us and each of these uses a different set of commands. The Archivers Control Panel takes the pain out of that and makes zipping and unzipping archives simple.
Yes you guessed it - it is another one of Thierry Godefroy's little gems and it is free!

System Requirements

Anyone can use this program provided they have a system with Toolkit 2 (Tk2) available, The Pointer Environment and enough disk space to the archiving files they need to use. I have put a basic system onto a DD disk and ran it on a Trump Card so there should be no difficulty there. If you want to be able to use all of the Archiving programs you will need to either have ED drives or be prepared to swap disks to get access to the programs although ACP itself is very small. Hard disk users, of course, have no problems.

What you get

The ACP zip file itself contains only three files:
ACP_HELP
ACP_OBJ
QLIB_RUN336mod
You will also need Richard Zidlicky's Signal extensions (sigext30_rext) to be LRESPR'd if you want to make use of the latest version of INFOZIP.
ACP_HELP is a text file which can be viewed in any of the normal text programs such as QD, EDITOR, QUILL etc and contains all the instructions you will need to run and use this program. The same file is called upon when you click on the '?' symbol in ACP and it opens in a separate window so you can view the help file whilst working. It is not in HTML format so you cannot jump from subject to subject but it is short and to the point. I would advise that you print it out if you are new to the program since it does give a

very good guide to using it.
ACP_OBJ is the program itself. This is a basic program which has been compiled using the QLIBERATOR compiler. It was created using Albin Hessler's Easyptr toolkit.
QLIB_RUN336mod are the Qliberator runtimes. These are needed to get the program to run and should be LRESPR'd before starting the program.

You should also have the Pointer environment files (PTR_GEN, WMAN, and HOT_REXT or SMSQ/E) running. It does make use of Jochen Merz's menu extensions (MENU_REXT) so load that if you have it. If you do not then the program will still run but you will have to select files by typing them in by hand.

Getting Started.

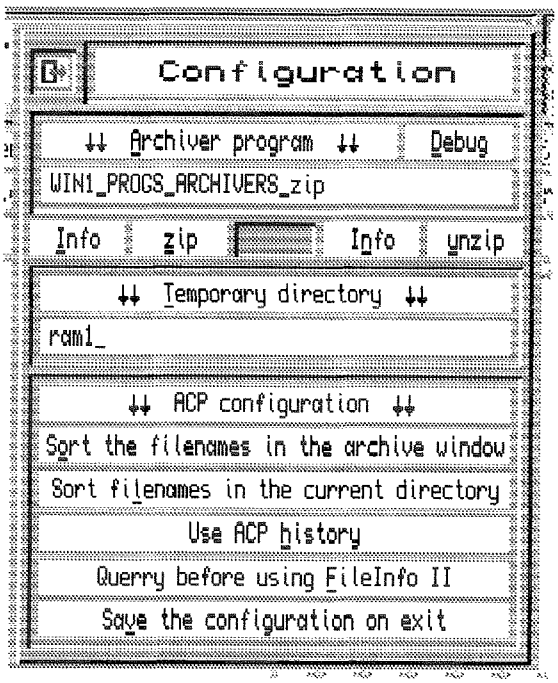
OK first problem. You have ACP_ZIP and you have UNZIP_ZIP how do you get started? Well if you have UNZIP540.bin you can just LRESPR the file at the command line and then follow the prompts. There will be a pause while it extracts itself to start and then a box will appear which asks for the directory for temporary files (I suggest RAM1_) and then a directory to extract to (I suggest RAM2_). The reason for using the RAM disks for this is that if anything goes haywire during the extraction you stand little risk of corrupting your hard drives File Allocation Table and you can easily reset the system and clear out the unwanted bits having extracted the files and copied them to disk. It will then ask you to run a short basic file called SFX_BAS. Do this and it will extract the whole archive to RAM2_. This will give you the UNZIP program that you can use to extract ACP itself. I did experience some problems doing this under SMSQ/E v 2.91 on the Super Gold Card but it extracted fine with standard QDOS / Minerva.
Once you have unzip on a disk (assume flp1_) you can copy the ACP_ZIP file to it and then use the command:

```
EX FLP1_UNZIP ;'FLP1_ACP_ZIP -d RAM1_'
```

This should extract the files to RAM1_ and you can copy them from there to the final destination that you will use the program from.
If you have already LRESPR'd the Pointer Environment, MENU_REXT and the Qliberator Runtimes all you need to do is EXEC the program and off you go.

Configure It

Before we can get started, however, we must first configure the program for our own system. I have created a subdirectory called Archivers and in there are all of the compression and de-compression tools. ACP can be configured by using the standard CONFIG program supplied with most P.E. programs. It can also be configured by Menuconfig, the more sophisticated version provided with most of Jochen Merz's Software.

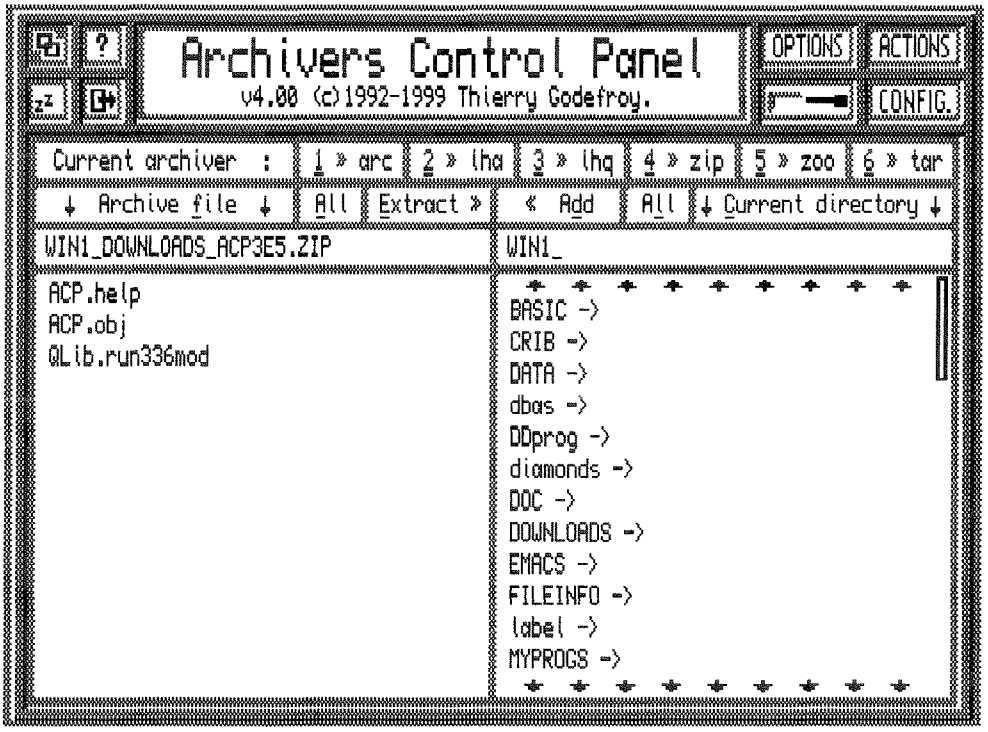


There are three config blocks to deal with. The first is directly involved with the program itself and contains 11 items. These will tell ACP where to find its help file, which is the default directory to look in for compressed files (I have created a subdirectory called downloads where I put all files downloaded from the BBS or other sources), which directory to start at for uncompressed files and a few other items like this. Among the items which are configurable here are whether the program should use FileInfo if present, whether ACP should retain its currently settings when you quit it, and which should be the default archiver.
The configuration of this part of ACP affects the way that you use it so apply some thought here. It is worth setting the program to save its config to the environment variables on exit because that means if you have to retrace you steps it will still point to the same directories. It is also worth turning on the sort routines because it is easier to find files which are sorted into alphabetical order. I would suggest that you set the default archiver to ZIP because that is the most common file format you will use. Setting it to use FileInfo II can be either a blessing or a curse depending on how you are using it and how you have set up FileInfo II. More of this later.
The second config block deals with the different archive formats and where to find their programs. this is very important because the program cannot use a particular compression program if it cannot find it.

The last block is the advanced settings. These allow you to let the archivers anticipate and fix program errors. Leave these turned on unless you find that they get it wrong.

The Program.

OK it is configured - let's go!



The first impression on seeing ACP is how similar it looks to the FileInfo II configurator. Thierry has used a similar colour scheme to that used on the Configurator so the two programs look very similar. The top left corner has three icons for **'MOVE'**, **'HELP'**, **'SLEEP'** and **'EXIT'** all familiar from FileInfo II. Then we have a program banner with the name of the program and the that of the author as well as the version number. The top right corner has a further four windows.

'OPTIONS' will call up a menu which will allow you to add extra settings to the command line. You can set it to do a few things such as overwrite files with same filename or force the result into QDOS format. The most important of these will be turned on already but you can experiment with them to get a feel for their uses.

'ACTIONS' comes next. This menu will allow you to allow you to perform tasks with the selected archive. These include checking the archive to ensure it is a valid file and getting the statistics on any or all of the files in the archive.

'CONFIG' is an 'on the fly' version of the second config block and allows you to make temporary changes to the programs configuration.

The last item is a drawing of a screwdriver. This represents Tools and allows actions specific to the BZIP, GZIP and compress formats. It will allow you to split large archives over several disks.

Beneath this is a line which is labelled 'Current Archiver'. This indicates the type compression you are using at the time. If you are extracting files from an archive this will affect the extension that the program looks for and which archiver it uses to display the directory of that archive.

Under this is another line of windows. the first has arrows pointing downwards and indicates that the window beneath it is the current archive. Next come a series of commands:

'ALL' will select all of the files in the current archive (if there is one)

'EXTRACT' well extract selected files in the current archive. You can select files in the 'Archive File' window by HITting them. If you 'DO' this without selecting any of the files in the window below it will first offer a window for you to enter the names of the files you want to extract. If you do not enter anything here it will then extract all of the files to the current default. If you 'DO' this without a filename in the Archive Window an error is generated.

'ADD' will add files to the selected archive. As with the 'EXTRACT' option if you highlight files in the 'Current Directory' window it will add them to the current 'Archive File', but unlike 'EXTRACT' it will not try to add all of the files in the 'Current Directory' window to an archive.

'ALL' Will allow you to do this.

Lets Archive!

At a basic level using Archivers Control Panel is simplicity itself. The next four windows in the display do all of the work. The two narrow windows display the file names that you are working with and the two larger windows beneath the files in the directory or archive.

To start with let us UNZIP a file. This process is very simple indeed especially so if you have

loaded the menu extensions. If you have set ZIP to be the default you can start right away with this month's cover disk.

Go to the long narrow window beneath the 'Archive file' label. You can either 'DO' (left click) the window and get a flashing cursor which will allow you to type in the file name, or 'HIT' (right click) it which will call up a menu_rext window and allow you to select the file from a list. Once you have selected the _zip file to be worked on the list of files in that archive will appear in the box below. If the box stays blank you have not loaded the signal extensions. If you want to extract all of the files in that archive you need only 'HIT' the window marked 'All' (or press 'a' on the keyboard). All of the files in this window will be highlit. If you only want to extract one or two of the files you can select them by going to the list window and 'HIT'ting the files you want to extract. These files will become highlit and they are ready for extraction.

If you want to view any of the files in the archive just 'DO' the file in the list and a window will appear displaying the file's contents. If you have other files highlit in the window they will be displayed one by one in the order they are listed in the window. This is useful if you have a README_txt file in it and you want to read the text before you extract the files in order to make sure that you only extract the files that you need. Just 'DO' the file name in the list and you can read it.

Once you have settled on the files you want to extract go to the narrow window on the right side of the program's screen. This indicates the destination or where the files should be unzipped to. Again you can 'HIT' and get the flashing cursor which allows you to enter the destination directory or device that you want to use. Far easier to let Menu_rext do the work and 'DO' it to get the 'Directory Select' which allows you to select the menu from there. If you press ESC while you are in this window The program will default to the DATA default directory. The directory which appears in this window by default is configurable in the config block so you can set it to RAM1_ and unzip to a RAM disk - just remember to copy the files to a non-volatile medium before you switch off. All that is left now is to click on 'Extract' and UNZIP will do its work.

As soon as the the archiver (in this case UNZIP) starts to work you will get a window appear which reports the progress of the extraction. If you wish to stop this at any time click on the 'GUN' icon in the top left corner and the process is terminated. The last file in the list may be incomplete so you should check this if you wish to keep it.

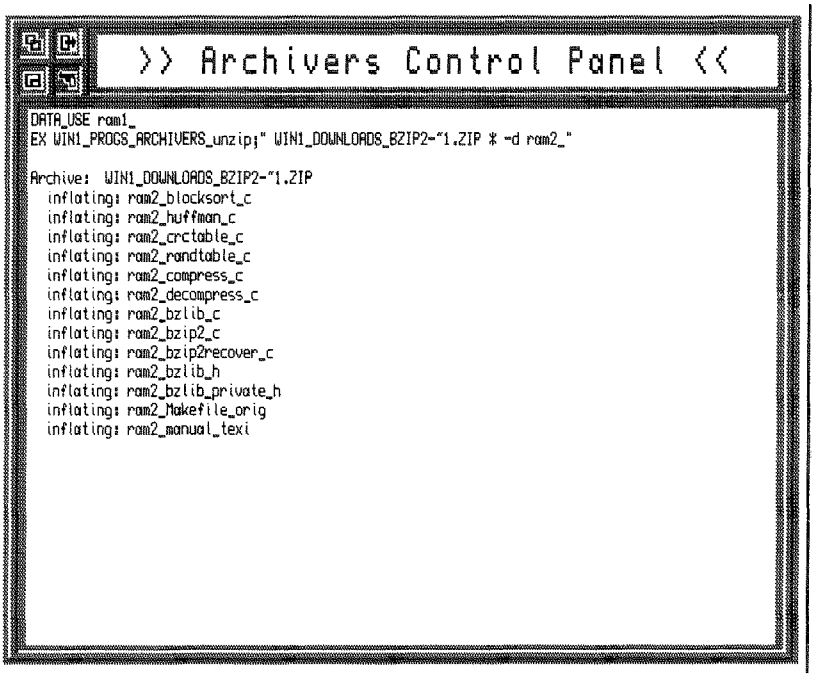
VICE VERSA

OK now let's do it the other way around. Maybe you want to zip up all of your letters into an archive so you can keep them on a floppy disk. 'DO' the narrow window on the right hand side and go to the directory where the files are or 'HIT' it and type in its name. You should now see an alphabetical list of the files in this directory. The sort routine will only sort the name and not the type of the file so you may find that any sub-directories within that directory are interspaced with your files. These are marked by the symbol '->' after the directory name.

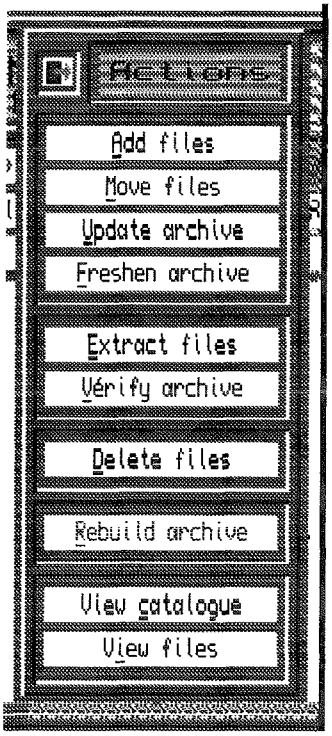
Again you can select all of the files in this window by clicking on the 'All' window or by pressing 'I'. You can also 'HIT' individual files as above to select files individually.

You now need to enter a file name for the resulting ZIP archive. Go to the left hand narrow window and 'HIT' it. When you get the flashing cursor enter the name of the file that you want to create. If you do not add '_zip' onto the name of the file ACP will do it for you. All you have to do now is to click on the 'Add' button and the 'ZIP' window will spring into life.

Once you have your Archive you may wish to add a new file to it. With ACP this is easy. Just



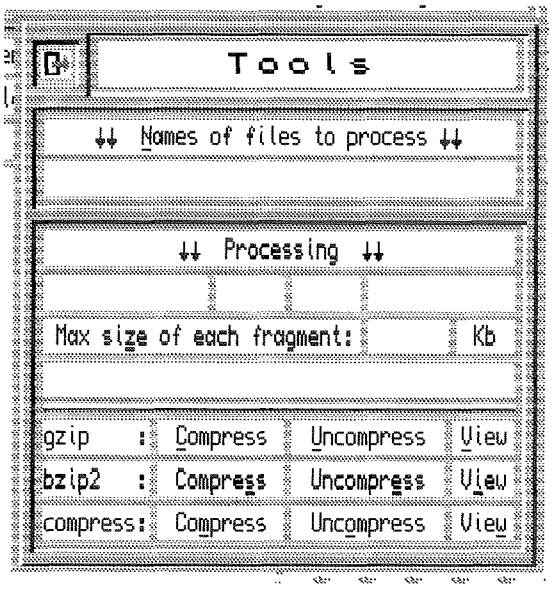
load the directory of the file you want to add, highlight the file or files that you need to add. Then go to the 'Archive File' window and call up the Archive that you want to add it to. You should see the list of files already there in the window below. Once this is set up just click 'Add' or press 'd' and the 'ZIP' program will appear and add the file to the archive.



This is where the FileInfo II compatibility comes in. Say you have a list of files that you want to archive but want to check a few things before making a final commitment. Just 'Do' the filename and FileInfo II will step in and perform the action that you have set FileInfo II to use. This is where the 'curse' part of the use of FileInfo comes in because if you have highlit several files and then accidentally 'DO'ne one of them FileInfo II will go on and do its work on all of the files.

There is an added function here because Thierry provides a set of BASIC routines which allow you to delete files directly from the ACP window. Very handy to tidy an archive before 'ZIP'ping. (You can also delete files from an archive by highlighting them in the Archive window and using the ACTIONS menu.)

Of course sometimes you wind up 'DO'ing when you should 'HIT' and FileInfo II gets launched when you did not intend it to be. Thierry has thought of this and added a config item that will ask you if you really meant to do this so you have a chance to abort the operation.



Not Only..... But Also

The Archivers Control Panel is a very useful and versatile tool and no QL user should be without it. It is simple to use and can be launched on almost any QL system. Not only does it allow you to extract and create ZIP files but it also deals with other, more obscure compression formats such as 'lha', 'lhq', 'arc', 'zoo', and 'tar'. Recent changes to the program allow it to use the GZIP and BZIP compressors so there is practically no format it cannot extract from.

Although the program itself is very simple to operate it has more complicated depths which the more advanced user can investigate and can fix errors generated by the common archivers. This has only been a brief dip into its use to get you started.

With 'FileInfo II' and 'The Archivers Control Panel' the QL has become a lot more powerful - amazingly enough they are both free too. I think all users should send a 'thank you' postcard to Thierry for such sterling work.

MonoCall

Nick Cheesman

The 1980's was a very exciting time for computer buffs like me. New machines coming out practically every month, each one better than it's predecessor. The Spectrum was my first computer and I enjoyed using it but there were always other machines to tempt the wallet. Sadly I was always strapped for cash and even when Dixon's started selling QL's at bargain prices, it was still beyond my pocket. Then before you could catch your breath, the 1980's were gone and the PC came to rule supreme. It also meant that the QL suddenly became very affordable on the secondhand market.

I bought my current batch of machines (four in all although one has a fault) several years ago for forty pounds complete with floppy drives, 512k of memory and a box of software that included Taskmaster, Flashback and Spellbound. I have since added a few more including Scrabble and the MSDOS emulator Solution. (I am keen to find a CP/M emulator if anyone has such a thing). All I need now is some QL World magazines and it'll be just like the old days again.

It was also about this time that I felt the need to publish a FanZine covering all manner of old 1980's kit now that official support had ended. I struggled for weeks to come up with a name that could be woven into various headings for each section (Historical, Nostalgia, Technical etc..) and after much torment came up with MonoCall (the FanZine would be printed on paper on a monochrome printer) with a programming style Call to each section from a menu. It never made it to print as the Internet suddenly appeared out of the blue and overnight it seemed that all other forms of publishing were obsolete particaularly as the need to spend money on paper (it was to be free), stamps and envelopes disappeared. So began the long tortuous path to learning how to write web pages. HTML was fine but it kept changing (and continues to do so) each time a new browser came out. I decided to standardise on two browsers (IE 3 and Netscape 4) for the PC and HTML 3 which included tables and frames which made the pages look more interesting even though many older browsers (including QL ones probably) couldn't view them. You can't please everyone I'm afraid but it would include users of PC's, Mac's, Amiga's and Linux. If I get enough complaints then I might put all the QL content on static plain pages but as I cannot run QL browsers on my lowly kit, it may be difficult to test it.

Just after making the decision to put MonoCall on the Internet, another major decision was made (apart from getting married and buying a house) which was to change my career at the tender age of 36. I had been in the Motor Trade man and boy first as a mechanic and then an administrator but redundancy forced a change of perspective as it so often does. The Motor Trade was (and is) unable to sustain me until I retired. Something else had to be found. The trick is not to avoid redundancy (which is probably impossible) but to be instantly re-employable should the worst occur.

I had of course been using computers for years and new a great deal but I also knew that I had much more to learn. If I was to become a Programmer (or anything else for that matter) then I would have to be really sure that I could go the course particularly if I was to give up my current career. I found another job in the Motor Trade but went to night school and obtained a GCSE in Computer Studies and a City and Guilds in C programming both of which proved very useful when I gave up work almost two years to the day after having been made redundant to start a two year full time course that culminated in an HND in Computing qualification and a large credit card bill (luckily my wife was earning at the time and we could just about live off her salary). Several months after having passed my final exams, I found a job as a Software Engineer and I have been with them ever since (two years).

I count myself lucky. It could have all gone horribly wrong. What if I had failed (many who were considerably younger than me were dropping out like flies). What if we got into debt and couldn't pay the mortage (I did get a grant but it was pitiful). What if I couldn't get a job. It all worked out in the end. Risks had to be taken and they paid off. I couldn't have done it without the support of my wife and not just financial.

Well that just about wraps it for the moment. My web site now has some QL content (with more to follow) as part of MonoCall Issue 4. Search for MonoCall in AltaVista and you'll find it. I also welcome feedback and cries for help on any computer or Internet related topics. Hope to hear from you.

The function of PROCedures and the procedure for writing FuNctions

Part 2: Why FuNctions? The result's the reason!

Mark Knight

A PROCedure replaces a traditional BASIC subroutine, and the PROCedure gets its name from the way it is intended to be used. PROCedures are defined whenever the programmer has a need for a particular procedure to be carried out in a clearly defined way by his program, especially if it will be needed repeatedly.

The FuNction gets its name from mathematics and from the way in which it is intended to be used. FuNctions in SuperBASIC and other BASICs are intended to be used whenever a calculation is required in a program, particularly if it is needed more than once. FuNctions in programming are named after mathematical functions, which are one of algebra's repeated processes.

The word "calculation" here should be seen in its computer programming context, not as a traditional arithmetic process. A FuNction may carry out string "calculations" and manipulations as well as, or instead of, numeric calculations. There are even reasons to write FuNctions that convert numbers into strings and vice-versa, in spite of the SuperBASIC interpreter's ability to do this for us a lot of the time.

Just like PROCedures, FuNctions can be set up to work on parameters and these work in exactly the same manner as PROCedure parameters, being set up in the DEFine line of the FuNction. Just like PROCedures, FuNctions can alter values of parameters and the results will vary depending upon whether the parameters are value or reference parameters.

Unlike a PROCedure though, we can PRINT the result of a FuNction or store it in a variable or an array. The result is passed back to the line that calls the FuNction using a RETurn line. As an example, a useful calculation in programming is elapsed time, perhaps in hours, minutes and seconds. A PROCedure could be written thus:

```
100 CLS
110 Elapsed_TIME 0,124124
120 Elapsed_TIME 124,333
130 Elapsed_TIME 0,1.234E8
140 :
150 DEFine PROCedure Elapsed_TIME(StartTime,
    EndTime)
160   LOCal Hours,Minutes,Seconds
170   LOCal Timer$(12)
180   Timer$=""
190   IF StartTime › EndTime THEN
200     PRINT"Start time after end time"
210     RETurn
220   END IF
230   Seconds = EndTime-StartTime
240   IF Seconds‹60 THEN
250     PRINT"00:00:";Seconds
260     RETurn
270   END IF
280   Minutes = INT(Seconds/60)
290   Seconds = Seconds - (Minutes*60)
300   Hours   = INT(Minutes/60)
310   IF Hours › 999999 THEN
```

```
320     PRINT"Out of range"
330     RETurn
340   END IF
350   Minutes = Minutes - (Hours*60)
360   Timer$  = Hours
370   IF LEN(Timer$)<2 THEN Timer$ = "0"&Timer$
380   Timer$=Timer$&":"
390   IF Minutes<10 THEN
400     Timer$=Timer$&"0"&(Minutes)
410   ELSE
420     Timer$=Timer$&(Minutes)
430   END IF
440   Timer$=Timer$&":"
450   IF Seconds<10 THEN
460     Timer$=Timer$&"0"&(Seconds)
470   ELSE
480     Timer$=Timer$&(Seconds)
490   END IF
500   PRINT Timer$
510 END DEFine Elapsed_TIME
520 :
```

This allows us to print the elapsed time any time (hee, hee) we wish, but it would be better if we could store the result and use it repeatedly, as well as printing it alongside other stored results. Storing the result alone could be acomplished by removing line 160 so that the Hours, Minutes and Seconds variables are still in existence after the PROCedure is called, but this doesn't solve the problem of storing more than one result. In addition, remember it is usually better to use LOCal statements to isolate the working of a PROCedure or a FuNction from the rest of a program, so we really want that LOCal to remain.
Instead of the above a much betterr solution is to DEFine a FuNction called something like "Elapsed_TIME$" which does the calculations for us, then gives us the result in a form we can store, PRINT or compare with others. The next listing shows one way of accomplishing this important job:

```
100 CLS
110 PRINT Elapsed_TIME$(0,124124)
120 PRINT Elapsed_TIME$(124,333)
130 PRINT Elapsed_TIME$(0,"2147483647")
140 FirstResult$=Elapsed_TIME$(0,12345)
150 SecondResult$=Elapsed_TIME$(12345,67890)
160 PRINT "First; ";FirstResult$, "Second;
";SecondResult$
170 :
180 DEFine FuNction
Elapsed_TIME$(StartTime,EndTime)
190   LOCal Hours,Minutes,Seconds
200   IF EndTime<StartTime THEN RETurn "Bad
Time"
210   Seconds = EndTime-StartTime
220   Minutes = INT(Seconds/60)
230   Seconds = Seconds-(Minutes*60)
240   Hours   = INT(Minutes/60)
250   Minutes = Minutes-(Hours*60)
260   RETurn Leading_CHAR$(Hours,2,"0")&":"
&Leading_CHAR$(Minutes,2,"0")&":"
&Leading_CHAR$(Seconds,2,"0")
270 END DEFine Elapsed_TIME$
280 :
290 DEFine FuNction
Leading_CHAR$(AnyNum$,Figures%,Lead$)
300   IF LEN(AnyNum$)<Figures% THEN
310     RETurn
FILL$(Lead$,Figures%-LEN(AnyNum$))&AnyNum$
320   ELSE
330     RETurn AnyNum$
340   END IF
350 END DEFine Leading_CHAR$
360 :
```

OK now we need to look at this listing carefully to see how it works, because clearly we have something very different to a PROCedure. In lines 180 and 290 the definitions look very smilar to PROCedure definitions, and our Elapsed_TIME$ FuNction in fact takes the same parameters as the Elapsed_TIME PROCedure we used earlier. Because it gives a string result though we have given it a "$" sign at the end of its name, always a sensible idea in both SuperBASIC and SBASIC. Line 190 is also familiar, keeping the variables "Hours", "Minutes" and "Seconds" isolated inside our FuNction, so we know they won't interfere with any similarly named variables our program might use if it grows later on.
Line 200 though, is very different to the equivalent lines in a PROCedure, because the RETurn is followed by the result string, in this case effectively an error message. All FuNctions must return results, they can't just return without passing anything back to the calling routine. If the desired calculation can't be carried out then it is desirable to let the calling routine know in some way, and I have chosen to return an error message here instead of the usual time in a string.
Lines 210 to 250 carry out the simple calculations to work out the hours, minutes and seconds in the result, and line 260 calls the later FuNction to put the result into a suitable string. Once it is ready the RETurn in line 260 sends it back to the part of the program that called the FuNction.
It is the first time we have shown this, but PROCedures may call other PROCedures or FuNctions and FuNctions may call other FuNctions, and PROCedures too if desired. Instead of repeating the string "calculations" as the PROCedure did in lines 360 to 490 we call a separate FuNction three times in line 260. This has the added advantage that any routines we write later in our program that need to pad a string to a minimum length with leading zeroes (or leading with spaces or anything else) can also call Leading_CHAR$.
If line 130 puzzles you then it is simply explained: SuperBASIC is capable of using ten figure in-

teger numbers within the range -2,147,483,648 to 2,147,483,647 but often LIST, SAVE or PRINT will give only seven figures. To preserve accuracy the long integer is placed in a string, so the interpreter will calculate the numeric value each time the program is run, preserving all ten digits. This allows our FuNction to work with up to 596,523 hours, 14 minutes and 7 seconds, which I think is a pretty useful working range.

Beware of bigger numbers if you use this FuNction in your own programs, as although the FuNction will still return a result it is likely to be inaccurate.

More seriously, let's look at some FuNctions which could be useful in many programs; one to convert a string to upper case, another to lower case and a third to capitalise a string:

```
100 CLS
110 Test$  = "Hello, this is a TEST string."
120 Test2$ = "abcdefghijklmnopqrstuvwxyz œ ç é
    ø 0123456789"
130 PRINT Test$\Test2$
140 PRINT UPPer$(Test$) \ LOWer$(Test$)
150 PRINT UPPer$(Test2$) \ LOWer$(Test2$)
160 PRINT CAPitalised$("some string or other
    originally entirely in lower case.")
170 :
180 DEFine FuNction UPPer$(Any$)
190   LOCal Count,Found
200   IF Any$="" THEN RETurn ""
210   FOR Count=1 TO LEN(Any$)
220     Found=CODE(Any$(Count))
230     SELect ON Found
240       =97 TO 122,128 TO 139
250         Any$(Count)=CHR$(Found ^^ 32)
260     END SELect
270   END FOR Count
280   RETurn Any$
290 END DEFine UPPer$
300 :
310 DEFine FuNction LOWer$(Any$)
320   LOCal Count,Found
330   IF Any$="" THEN RETurn ""
340   FOR Count=1 TO LEN(Any$)
350     Found=CODE(Any$(Count))
360     SELect ON Found
370       =65 TO 90,160 TO 171
380         Any$(Count)=CHR$(Found ^^ 32)
390     END SELect
400   END FOR Count
410   RETurn Any$
420 END DEFine LOWer$
430 :
440 DEFine FuNction CAPitalised$(Any$)
450   IF Any$="" THEN RETurn ""
460   Any$(1)=UPPer$(Any$(1))
470   RETurn Any$
480 END DEFine CAPitalised$
490 :
```

Run the program to see the effect of the FuNctions, and look at the listing to see how they work. The important part here is that they return results using the SuperBASIC RETurn in suitable lines. Just looking at UPPer$, examine line 200 which prevents error messages by returning an empty string right at the start. The FOR loop defined in lines 210 to 270 steps through the string, converting any suitable characters to upper case. Once this is done, if anything has been done at all, line 280 RETurns the result.

Another simple example uses the Turbo toolkit function SYS_VARS, which not surprisingly returns the base address of the system variables. Although it might be considered bad practise to POKE into the system, it is legitimate to use the various PEEK functions to read them. One use might be this:

```
100 PRINT "System free memory=";Free_SPACE
110 :
120 DEFine FuNction System_SPACE
130   RETurn PEEK_L(SYS_VARS + 16)-
      PEEK_L(SYS_VARS + 12)
140 END DEFine System_SPACE
```

This is very different to the previous examples because it doesn't do some work and then RETurn the result; it does both in a single line. FuNctions may RETurn the values of expressions as well as single values.

So the concluson of the preceding two sections is that we use PROCedures when some action is required and FuNctions when a result is required; that is what these two structures are intended for. The only way to learn is to keep using them in your programs and collect and examine other programmers efforts whenever possible.

You and Your Software - Just good Friends?

Part 7 - Manuals - the great unread.

Geoff Wicks

Deep in the archives of Just Words! is a lengthy letter from a user giving a detailed criticism of one of my programs. It is a letter I valued, and I used some of his suggestions when I upgraded the program. Nevertheless, I could tell from the content that he had not read the program's manual. Indeed, I could put it even stronger. There is evidence he had not even seen the manual, and was using a pirate version of the program.

Manuals are a major headache

for the software author, particularly in the commercial sector. They are costly and time consuming to produce, and become rapidly out of date as a program is developed and improved. In spite of his efforts the software author knows that many users of his program will give the manual little more than a cursory glance.

Many people argue that a good computer program does not need a manual. If software is well written, you should be able to use it intuitively and help will be provided on-screen to get you over the difficult bits. It is a powerful argument. When I test or review software, I initially keep the manual firmly shut to see how far I can use the program without looking at it. It is a good test of a program's user friendliness.

If we want to know what a world without manuals would be like, we need look no further than the PC. Walk into practically any bookshop and you will see shelves full of expensive guides to the more popular PC programs. Modern PC software rarely comes with a comprehensive manual, and has extensive on-screen help, but this has its limitations. Many users want more information than can be provided on-screen, and in effect have to pay £15 to £20 extra to buy a manual for the program.

The QL is too small to spawn a cottage industry of alternative manuals, and the software author has to write his own. He could try to write an on-screen manual, but would find this would probably more than double the program's memory needs. Whatever he does, the manual will be a lot of effort for little reward. However good or bad a manual is, most users will probably not read it.

There are times when software authors are tempted to jump on a table at a QL Show, and scream at the top of their voices "RTM" ("read the manual") or even "RTFM". This is a temptation to be avoided, because it is quite reasonable that many users do not read the manual. After all, why do you buy software? To plough through pages of technical information? To read through a detailed history of the program's many versions? To be confronted with a lengthy discussion about the background and theory? Or do you buy it to get the program up and running on your machine as quickly as possible?

If you want your users to read your manuals, then it is your job to give them something they will want to read.

An impossible task? This is what a QL Today reviewer wrote about a manual:

"It's laid out logically and allows you to enter it at different levels and stages. So it's easy to find the bit you need if you get stuck. Or it's easy to blitz through the Quick Start section if you can't wait to get your hands on the program. Or it's easy to quickly go to the Customisation section if you don't like the results with the current configuration.

Essentially it's difficult to fault. It told me everything I needed to know, gave me no difficulties in finding the relevant section, explained the menu structure concisely, gave numerous examples in the glossary and covered everything."

At this stage I have to confess this a bit of free advertising, because the reviewer was describing a Just Words! manual. So what are my concepts when I write a manual?

Firstly design is important. A manual must look attractive, and be laid out in such a way that the reader can quickly find what he is looking for. Just Words! manuals are written, spell-checked and style-checked on the QL, but are then transferred to a PC word processor for the final design. Sorry about that, but QL word processors, with the possible exception of Text 87, have fewer design possibilities than PC programs.

The present generation of Just Words! manuals are A4 folders, with two columns of text, which is easier to read than one column. There is a bold header at the top of each page describing its contents. Within the text, important items such as menu commands are highlighted by being printed in bold capital letters in a slightly larger font than the main body of the text. When a user thumbs through the manual, these items stand out, which makes it easier for him to find what he is looking for. If he cannot find it, there is an index at the back to help him, and there are also illustrations of the main menus.

If you are writing for the freeware market, your manual will be either a Quill or ASCII file, and you will have fewer design possibilities, but it is worthwhile to look at the layout of your document to ensure it is as user-friendly as possible.

Secondly a Just Words! manual has three main sections, designed for different stages of experience in using the program.

The first section is Basic Information. It is what you need before you buy the program and during installation. It gives information people need when they are browsing through the manual at a show. It tells them what the program does, what the system requirements are, what files are on the disk and how to install the program on a hard disk.

The second section is the Quick Start. This has proved to

be the most popular feature of my manuals. It is a simple step by step tutorial to illustrate the main features of the program, and can usually be followed without further reference to the manual.
The third section is the remainder of the manual. Some users will read this, and some will not. If a user wants to read the manual from cover to cover, it will tell him about the program in some depth. Alternatively if he does not want to read it, he can use it as a reference work for when he gets stuck. Hence the importance of a design that makes things easy to find.
Manuals are the great unread, but some are more unread than others. A user-friendly manual is an essential part of user-friendly software.
Next time: What the user doesn't see.

CAUTION: HOTKEYS - Don't Burn Your Fingers, Part 3

David Denham

So far in this series we've covered most of the BASIC commands and functions used in the Hotkey system. In this part we'll look at two general extensions not fully covered so far (HOT_KEY and HOT_WAKE) then take a look at the Stuffer Buffer and its associated controls in the next issue.

HOT_KEY

If you have used Toolkit 2's ALTKEY command to define keys to hold shortcut strings for you, you'll find HOT_KEY a little familiar. HOT_KEY is a function rather than a command like ALTKEY. In QDOS terms it works rather differently - the QPAC2 manual explains that ALTKEY works via a polling task while HOT_KEY is via a hotkey job. As far as you the user are concerned you can achieve pretty much the same results with both. In essence, HOT_KEY assigns text to a key. That's it.
In case you wonder what makes this any different to HOT_CMD, bear in mind that HOT_CMD is used to send commands to BASIC, rather than text as such. HOT_KEY sends the text into the current keyboard queue instead.

A simple example

Suppose the period '.' key is broken and while we wait for a new keyboard membrane from Tony Firshman we decide we could work around this problem by defining an alternative keypress which allows us to press ALT and another key to give us a period character instead. In this case, we'll make ALT comma key give us a period or full stop character.

```
ERT HOT_KEY(",",".")
```

The confusing looking strip of quotes and commas hides a simple fact, that when ALT and comma keys are pressed, the QL gives us the full stop symbol we needed to work around our problem.
Another example. Suppose the QL is used in a business which exports goods to various countries. You frequently need to access the currency symbols not marked on your keyboard but available via CTRL/SHIFT keys, as listed in the Character Set section of the Concepts User Guide. We can define hotkeys to give us these currency symbols from easily memorable keys (ALT p for Pound Sterling, ALT y for Yen, ALT d for Dollar, ALT e for Euro if you have SMSQ/E and ALT c for Cent)
We can examine the character set with this short program which tells us which CODE value represents each character:

```
FOR a = 32 TO 191
PRINT ! a&'='&CHR$(a) !
END FOR a
```

On a U.K. character set machine we'd get the following codes:
Pound = 96
Dollar = 36
Cent = 157
Yen = 158
Euro = 181 (recent versions of SMSQE only)
So we can define easily memorisable currency symbol keys:

```
ERT HOT_KEY('p',CHR$(96))
ERT HOT_KEY('d',CHR$(36))
ERT HOT_KEY('c',CHR$(157))
ERT HOT_KEY('y',CHR$(158))
ERT HOT_KEY('e',CHR$(181))
```

You can use CHR$ to add non-printable codes too, such as the codes for the function keys, e.g. this example defines ALT f as the F3 key, not very useful in itself but becomes more useful when you combine this with the letters that control the commands menus in Quill, Archive, etc. 240 is the character code for the F3 key (these are listed in the Character Set section of the Concepts part of the QL User Guide)

```
ERT HOT_KEY('f',CHR$(240))
```

We can create a little macro from this to create a shortcut to the PRINT command in Quill. Normally, the sequence of key presses would be:

QUANTA

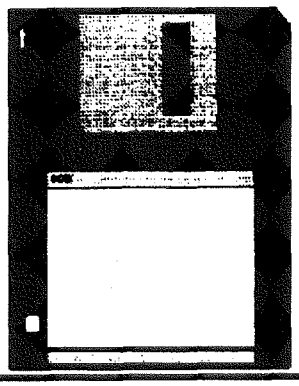
Independent QL Users Group

Worldwide Membership is by subscription
only, and offers the following benefits:
Monthly Newsletter - up to 40 pages
Massive Software Library - All Free !
Free Helpline and Workshops
Regional Sub-Groups. One near you?
Advice on Software and Hardware problems
Subscription just £14 for UK members
Overseas subscription £17

Barclaycard: Visa: Access: Mastercard

*** Now in our SEVENTEENTH successful year ***

Further details from the Membership Secretary



Bill Newell
213 Manor Road
Benfleet
Essex
SS7 4JD
Tel. (01268) 754407

press F3 for the command menu press p for Print press ENTER when it asks "Current" press ENTER again when it asks "Whole" press ENTER again when it asks "to printer"
So we need an F3 keypress code, a "p" and three ENTER characters.
It is worth noting that HOT_KEY has a special feature when dealing with ENTER characters and multiple strings. If you include several strings in a definition separated by commas, they are sent with an enter character between them. If you add an empty string after a comma (a "") in effect this creates a blank line - useful right at the end of a definition for example.
So to define ALT f as our Print whole document shortcut in Quill we use:

```
ERT HOT_KEY("f",chr$(240)&"p","","","")
```

Bear in mind that the apostrophe and quote symbols are interchangeable in the hotkey system - you could use '' or ""
This feature allows us to define keys which hold several lines of text. We can use this to enter commonly used addresses for example:

```
ERT HOT_KEY('j','Jochen Merz','Im Stillen
Winkel 12','D-47169 Duisburg 11')
```

This inserts the address as 3 lines of text. Inserted into Quill, these appear at the left margin, which is no good for a letter. They would need to be manually TABbed over to the right. No problem, get the hotkey to do it using CHR$(9) as TAB key presses:

```
ERT HOT_KEY('j',CHR$(9) &CHR$(9) &CHR$(9)
&'Jochen Merz', CHR$(9) &CHR$(9) &CHR$(9)
&'Im Stillen Winkel 12', CHR$(9) &CHR$(9)
&CHR$(9) &'D-47169 Duisburg 11')
```

Although it would involve some long and clumsy key definitions, you could store your letterheads as little text macros in this way - it wouldn't take too much effort to write a short BASIC program which let you type in a few lines of text (a few INPUT statements), asked you which key to assign these definitions to, and then build a HOT_KEY statement by printing it to a file with a line number in front of it which you could merge with your boot program!

HOT_WAKE

A HOT_WAKE is rather like a HOT_PICK in the sense that it will pick the job specified, but in addition it sends a WAKE event to the job which in the case of some programs such as QPAC2 menus which respond to WAKE events causes a particular action to take place, such as refresh a menu. The Files menu of QPAC2 will read the disk again to update or refresh its list of files upon being given a WAKE signal.
In addition to this action, if the specified job is not running, but an executable thing of that name exists it will be started.
If you don't know much about Things (and I certainly don't) just think of executable things as little programs which have been stored in memory but are not actually running - they are like BASIC extensions you load with RESPR or LRESPR, they only actually run when you type in their name. These executable Things can be started with an EXEP command usually rather than an EXEC command. Examples of executable Things are QPAC2 menus (Files menu, Jobs menu, Exec menu etc) and most normal Job style programs (those you can EXEC from disk) which have been installed in memory with HOT_RES and HOT_CHP commands. In effect, you have used HOT_RES or HOT_CHP to plant them ready in memory, then you use HOT_WAKE commands to start them and pick them. In addition, there is also the HOT_THING function which allows you to start such Things, just to complicate matters, so I'll leave it at that as I am getting into deep water over Things.
One particular feature is that HOT_WAKE tries to pick a job name, if that doesn't exist it tries to start a job instead. In effect this means that it only ever starts one copy of a named job. This is useful for QPAC2 Files menus - if you already have one open it can be useful that the HOT_WAKE first tries to use the existing one before cluttering up your screen starting multiple copies!

```
ERT HOT_WAKE('f','Files')
```

This will do one of two things - if a QPAC2 Files menu is already open it will pick that menu and send a wake event to update the list of files as though you had pressed CTRL F2 or hit the little lightning icon top right of the menu. But if no Files menu was running when you pressed ALT f a new copy is run in the normal way.

HOT_THING

```
ERT HOT_THING('f','files')
```

will start an executable Thing called Files, in this case the QPAC2 files menu, and unlike HOT_WAKE always starts a new copy - useful if you want to use two files menus to compare lists of files on separate drives. Perhaps more usefully, you can set up a HOT_THING hotkey to start the EXEC menu of QPAC2 from which you can access most of the Things in QPAC2:

```
ERT HOT_THING('x','Exec')
```

QPAC2 has a useful Thing called BUTTON_SLEEP which minimises programs down to a button in the button frame. It is possible

to access this from the EXEC menu, but why bother when you can simply assign it to a HOT_THING hotkey:

```
ERT HOT_THING('s','Button_Sleep')
```

So when you press ALT s the current program gets made into a sleeping button in the button frame, clearing precious screen space and reducing clutter. This assumes of course it's a program which can be made into a button (most normal programs can).

A dozen Progs You shouldn't live without

Darren Branagh

In a recent issue of QUANTA, (Feb. 2000) John Gregory, took a look at some of the programs he found useful in the QUANTA library over the years, and has installed on his system. I found it a very informative article, and thought QL Today could benefit from the same kind of thing. I was aware of all of the programs John mentioned, (even having written one of them myself!) but had forgotten about a few of them until reminded by the article.

The QUANTA library really is a treasure trove of programs, many of which cannot be found elsewhere, and in my opinion membership of QUANTA is justified on this alone. However, there are also quite a lot of other PD and freeware programs being supplied by the PD libraries, most notably Phil Jordan.

In the following paragraphs, I've taken a look at some of the programs I use both regularly, and now and again when needed. Together, they make up a valuable arsenal of programming power, and the best part is, they didn't cost me a penny!! These programs are so simple to run and use that they don't warrant a review of their own, as they are simply a case of loading them and they do their thing then go away. I know I couldn't do without some of them most of the time, and maybe you are unaware of them, and may find them useful too.

1. THE STRIPPER

For all of those readers who think a stripper is something else entirely, shame on you. It's actually a simple to use program to remove unwanted control codes from files, in order to turn them into plain ASCII text. It works perfectly on QUILL doc files for example. All you do is load it, and type the name of the file to strip, and it does it. easy and very useful.

2. SCR2BMP

Written by Norman Dunbar, This program will convert standard QL screens (32K) into bitmap images, which are common on PC's (The microsoft windows wallpaper is stored in bitmap format, for example) very useful again, and as easy to use as THE STRIPPER - just EXEC it, and give a _SCR filename and location, and it will output it as a Bitmap screen, which can be saved where you like.

3. SCR2PCX

As above, but will convert to the equally popular PCX graphics format instead. There are tonnes of PCX Clipart in the PC world, and this allows you to add some more. This one can handle mode 4 or 8 screens, with parameters being passed at startup, and even supports wildcards too. eg.

```
EXEC SCR2PCX;'flp1_*_SCR 8
ram1_'
```

will convert all the screen files on flp1_ in mode 8 to PCX files on ram1_.

4. THE COPIER

Have you ever had problems copying a disk with level 2 subdirectories on it? I remember I did, when I first moved from a Trump Card to a QXL in one go, as this was the first piece of hardware I had which supported it. That was, before Dilwyn Jones wrote The Copier. This excellent program will make an exact copy of any disk, including ALL subdirectories, and will copy all files to the correct location. You can even specify which files or directories are copied. The locations can be any device, so it is even a rudimentary and basic way of doing backups of your hard drive. All you do is enter the drive or directory to copy from, and the destination - simple. Another Program called SubCOPY does the same thing, and there is also a utility on the ProWesS disk which will also do this (ProWesS is now no longer commercial software).

5. QFORMAT

Again by Norman Dunbar, this is a quick disk formatter. It will completely erase any DD, HD or even ED disk, with a new volume name of your choice, in only a few seconds. Much quicker than formatting from the command line under Superbasic, which takes several minutes.

6. SORTBACK

Another program by Dilwyn, which is a simple way of alphabetically sorting all the files in your disks. It will sort the files in any order, and allows you to copy them back onto

the disk in the way they where sorted. Makes it very easy to find certain files on your disks if all the files appear alphabetically when DIR'ed, I can tell you!!

7. QRISTEN

Pronounced "Chris-en" (You'll see why in a minute) this is a pointer program that allows you to quickly rename a disk without the need to format it first. Useful if you have copied the contents of something onto a disk that has a meaningless volume name on it. I wrote a Superbasic program some time ago to do this, but this is much easier to use and can be placed on a Button easily too. There is another similar program called simply labeller, which does the same job - Both are PD.

8. FREEMEM

This displays the amount of free memory on your system, on a button in your button frame. Simple and useful, and updates constantly. Available for download from Jonathan Hudson's website.

9. DOCVIEWER

A pointer driven Quill doc viewer, quite simply put. Fire this up and you can have a quick read of any quill file easily, with the added bonus of being able to scroll through it all using just the mouse, which you can't do under Quill, or Xchange. All the original formatting of the quill file is displayed correctly, although its worth pointing out that changes cannot be made to files as its simply a viewer program.

10. HTML MACHINE

A clever program written by Roy Wood to allow HTML (hyper Text Markup Language) which is common on the Internet to be typed on a QL. The HTML Machine will pop-up over a text editor (it's designed for use on QD, but should work with any) and allow you to select from a range of HTML tags to be placed into the text. As HTML grows in popularity, this is an excellent program to aid using basic HTML on a QL.

11. GRAPHICS VIEWER

Another one by Dilwyn (does this man ever take time to breathe?) and the title says it all. It's a great little program, which can be used to view files in a wide variety of formats - _PIC and _SCR are supported, QL DTP programs pages and even a couple of PC formats. Fully pointer driven, with resizable windowing and the pointer file-select menu to choose your clipart or graphic to view. As easy to use as all Dilwyn's programs.

12. PROCMAN

Ever written a really great set of superbasic procedures for a certain project, only to find the same procedures useful in another project some time later?? Rather than SAVE and MERGE files to get at them, PROCMAN will easily seek out all the procedures in any superbasic program, and allow you to extract them individually or all together, as you wish, and save them for later use. A gift for the budding and experienced programmer alike.

There you go. All of these are in the QUANTA library, and also most are available from Phil Jordans Library service too. If you are not a QUANTA member, there are many programs in there that you will not be able to get, and it's this reason alone that justifies the membership price in my mind. The QUANTA library is full of goodies, so order a copy of the LIBGUIDE on disk today, you'll be glad you did. If you're not a member, then Join - as you'll find it is the best few quid you've ever spent.

QLTdis - part four

Norman Dunbar

Following on from PRINT_HEX, we have some useful routines to copy strings, append one string to another, and to add a byte, word or long word of text to an existing string.

First up, is STR_COPY which simply takes the string at A1 and copies it to A2 overwriting whatever was at A2 originally. Very similar to LET A2$ = A1$ in SuperBasic. Here it is:

```
* ========================================================
* str_copy :
* ========================================================
* Copies a string from A1.L to A2.L overwriting the
* current contents of the buffer at A2.L.
* No errors returned. If the length of the string to
* be copied is even, move words - it is quicker.
*
* Expects A1.L to hold the 'from' string address.
* Expects A2.L to hold the 'to' string address.
* Corrupts D4, A1 and A2.
* ========================================================
str_copy
        move.w  (a1)+,d4    ; Get size of 'from' string
        move.w  d4,(a2)+    ; Set new size of 'to' string
        btst    #0,d4       ; Is D4 even ? (size of string)
        bne.s   str_next    ; Do byte moves - odd sized string
```

```
str_even
        lsr.w   #1,d4       ; Divide length by 2
        bra.s   str_enext   ; Do word moves - even sized string

str_emove
        move.w  (a1)+,(a2)+ ; Move a single word
str_enext
        dbra    d4,str_emove ; And the rest
        rts

str_move
        move.b  (a1)+,(a2)+ ; Move a single byte
str_next
        dbra    d4,str_move ; And the rest
        rts                 ; Exit
```

As you now know, a QDOS string is held in memory starting at an even address, with a single word of data holding the length of the string (not including the word itself). Immediately after this word is the individual bytes of the string.
To copy one string to another is quite simple - get the size of the 'from' string and move it to the 'to' string's length. Then copy each byte across.

In this routine we first get the length of the string we are copying into D4.W - we shall be using this as a counter of bytes to copy. This is then stored at the destination address held in A2 and A2 is set to pint to the position where the first character will be stored when we start copying.

If bit zero of the length of the string is clear then we have an even number of bytes to transfer. This means that we can use word sized copies instead of byte sized and thus do fewer of them which speeds up the program a bit. Obviously if moving words around, we only need half as many, so D4 is divided by 2 to keep the count correct.

If the length is odd we simply copy each byte over one by one. This keeps the routine fairly simple as we could have easily moved the first 'n' words and when only one byte left moved it on its own.

As an exercise, try rewriting str_copy to do the following:

1. Calculate how many words to move and how many spare bytes (either 1 or 0)
2. Move the required number of words first - guess why?
3. Move the final byte.

In case you were wondering, if you move the spare byte first, this makes the to and from addresses ODD so the first word you move will cause an address exception error - oops!

The next routine allows us to append to a string. Appending to a string is a bit more difficult. LET A2$ = A2$ + A1$ is the equivalent Superbasic code. Lets see how this is done.

```
* ======================================================
* str_append :
* ======================================================
* Appends a string from A1.L to A2.L adding it to the end
* of the current contents of the buffer at A2.L.
* No errors returned.
*
* Expects A1.L to hold the 'from' string address.
* Expects A2.L to hold the 'to' string address.
* Corrupts D4, D3, A1 and A2.
* ======================================================
str_append
        move.w  (a1)+,d4    ; Size of 'from' string
        move.w  (a2),d3     ; Size of 'to' string
        add.w   d4,(a2)+    ; New size of 'to' string
        adda.w  d3,a2       ; New 'to' string end position
        btst    #0,d4       ; Is the from string even sized?
        beq.s   str_even    ; Move string in words
        bra.s   str_next    ; Copy all bytes over
```

If this seems very short - it is!

All we do here is collect the 'from' string length and the 'to' string length. Having done this, the size of the 'to' string has to be the combined size of both strings, so we add the sizes together and store in the size word for the 'to' string. Now we have the to string set to the correct length, we copy the bytes of the 'from' string over to the end of the 'to' string. To calculate this position, we add the old size word of the 'to' string to the address of its first character.

Assume the two strings are as follows and that the ADDR values are simply the address relative to the start of the two strings as opposed to an actual address in memory otherwise both strings would be stored in the same place!

```
Addr : 00 01 02 03 04 05 06 07 08 09 10
From : 00 04 F  R  O  M
To   : 00 02 T  O
```

We add the sizes together to get this:

```
Addr : 00 01 02 03 04 05 06 07 08 09 10
From : 00 04 F  R  O  M
To   : 00 06 T  O
```

A2 is adjusted to point at the first character (the 'T') in the 'to' string by the 'ADD.W D4,(A2)+'. Note that you cannot ADDA.W D3,(A2)+ which is why there are two different instructions in this routine to add to the address registers.

So if we now have A2 holding address $02 in the 'to' string and add the length of the old 'to' string -

which was $02 to it, we get A2 pointing at address $04. Looking ath the above, $04 is the place where we want to put the first character from the 'from' string. Easy when you see it! (I use a lot of squared paper when I am working with addresses!)

So, once we know this address, we skip backwards into STR_COPY to copy the bytes over either as words or as bytes. Code re-use - what a wonderful thing!

At various places in the final program, there will be a need to add one, two, three or four characters to an existing string. This is simply adding characters and not proper strings so the above routine, STR_APPEND, cannot be used as it is, because we don't have a word at the start of the characters we wish to add.

It would be quite simple to do something like the following (assume that A5 holds the address of the first spare character space after the END of the string to be appended to)

```
MOVE.B  #'L',D4
MOVE.B  D4,(A5)+
```

And this will actually work without problems all the time. However, the following will not work all the time - why?

```
MOVE.W  #'A7',D4
MOVE.W  D4,(A5)+
```

or

```
MOVE.L  #'ADDA',D4
MOVE.L  D4,(A5)+
```

Have you figured out what is wrong ? It is quite simple, and quite difficult to figure out when testing a program that seems to randomly cause problems. If we start off with A5 holding an EVEN address, the above code will work. We are allowed to perform word and long sized operations at even addresses. But, if we have already added one byte to the string at A5, A5 now holds an ODD address. The above code will now cause an address error. Nasty, so beware.

The following routines allow the addition of four, two or one bytes to the end of the A5 buffer. These will be used when building up the decoded instructions in the main program.

```
* ========================================================
* str_add_l :
* ========================================================
* Adds the 4 bytes in D4.L to the buffer at A5.L. This is
* our decoded instruction ‹so far› and we are adding
* another 4 bytes to it. See text for why we do it this
* way and not as MOVE.L D4,(A5)+ !!
*
* Expects D4.L to hold the 4 bytes we wish to append to
* the buffer.
* Expects D6.W to hold the size decoded so far.
* Expects A5.L to point to the first free space in the buffer.
* Preserves D4,  A5 and D6 are updated.
* ========================================================
str_add_l
        bsr.s   a5_even       ; Test if A5 is even
        bne.s   str_add_4     ; A5 is odd
        move.l  d4,(a5)+      ; A5 is even - yippee !
        addq.w  #4,d6         ; Update counter
        rts                   ; Done

str_add_4
        swap    d4            ; Change 'ABCD' to 'CDAB'
        bsr.s   str_add_2     ; Append 'AB' part first
        swap    d4            ; Change 'CDAB' back to 'ABCD'
        bsr.s   str_add_2     ; Append 'CD' part.
        rts

* ========================================================
* str_add_w :
* ========================================================
* Adds the 2 bytes in D4.W to the buffer at A5.L. This is
* our decoded instruction ‹so far› and we are adding another
* 2 bytes to it. See text for why we do it this way and not
* as MOVE.W D4,(A5)+ !!
*
* Expects D4.W to hold the 2 bytes we wish to append to the
* buffer.
* Expects D6.W to hold the size decoded so far.
* Expects A5.L to point to the first free space in the buffer.
* Preserves D4,  A5 and D6 are updated.
* ========================================================
str_add_w
        bsr.s   a5_even       ; Is A5 even ?
        bne.s   str_add_2     ; Apparently not
        move.w  d4,(a5)+      ; It is - store D4.W in the buffer
        addq.w  #2,d5         ; Update counter
        rts                   ; Done

str_add_2
        rol.w   #8,d4         ; Change 'EF' to 'FE'
        bsr.s   str_add_b     ; Append the 'E' part first
        rol.w   #8,d4         ; Change 'FE' back to 'EF'
        bsr.s   str_add_b     ; Append the 'F' next
        rts                   ; Done

* ========================================================
* str_add_b :
* ========================================================
* Adds the byte in D4.B to the buffer at A5.L. This is our
* decoded instruction ‹so far› and we are adding another
* byte to it.
*
* Expects D4.B to hold the byte we wish to append to the
* buffer.
* Expects D6.W to hold the size decoded so far.
* Expects A5.L to point to the first free space in the buffer.
* Preserves D4, A5 and D6 are updated.
* ========================================================
str_add_b
        move.b  d4,(a5)+      ; Store a single byte
        addq.w  #1,d6         ; D6 holds size of text so far
        rts                   ; Go back from whence we came.
```

```
* ======================================================
* a5_even :
* ======================================================
* Tests bit zero of A5 to see if it is even. If it is we can
* store words and long words at the location held in A5.
* Otherwise we can't.
*
* Expects A5 to hold an address.
* Preserves everything.
* Returns Z set if A5 is even, clear if A5 is odd.
* ======================================================
a5_even
        exg     d0,a5           ; A5 is required in a data
register
        btst    #0,d0           ; Sets Z if D0 is (now) even
        exg     d0,a5           ; Doesn't affect the flags
        rts                     ; Done

* ======================================================
* str_add_3 :
* ======================================================
* Adds the 3 bytes in D4.L to the buffer at A5.L.
* This is our decoded instruction ‹so far› and we are
* adding another 3 bytes to it.
*
* Expects D4.L to hold the 3 bytes we wish to append to the
* buffer.
* Expects D6.W to hold the size decoded so far.
* Expects A5.L to point to the first free space in the buffer.
* Preserves D4,  A5 and D6 are updated.
* ======================================================
str_add_3
        swap    d4  ; Change '?ABC' to 'BC?A' (? = don't care)
        bsr.s   str_add_b       ; Append 'A' part first
        swap    d4              ; Change 'BC?A' back to '?ABC'
        bsr.s   str_add_w       ; Store the 'BC' part
        rts                     ; All done.
```

The easiest one to figure out is STR_ADD_B as it simply adds one byte to the string length. We will be using D6 to hold the completed size of the decoded instruction, so we increment its total as well.

The routine a5_even is called when we want to add 2 or 4 bytes to the A5 buffer. This simply returns the Z flag set if A5 is an even address. As you cannot test the individual bits in an address register, we swap A5 and D0 over, test D0 - which sets the Z flag accordingly, swap A5 and D0 back again - which does not change any of the flags, so Z is still correct and then we return.

Next up, is STR_ADD_W which adds two bytes, one at a time to the A5 buffer by calling STR_ADD_B twice. Notice how similar this is to the HEX routines earlier. The process is very similar. We must swap the order of the bytes in D4.W before we add them to the buffer. If we detected that A5 is even, then we don't have to bother with swapping etc and we simply store the word in D4 at the address in A5 - so much simpler when things are even.

And again, the same with STR_ADD_L where we must swap low and high words over before we can add text to the buffer. Again, if A5 proves to be even, we simply store the long word in D4 at the address in A5.

After STR_ADD_B there is STR_ADD_3 which is used to add three bytes from D4 to the buffer. This is very similar to the STR_ADD_L routine, but only stores the lowest 3 bytes of D4 in the buffer. It does this by first swapping the two words of D4, storing one byte froom the (now) low word, swapping the words back again and storing the entire low word. Simple stuff.

Finally for UTILS_ASM, we have the routine ADD_SPACES which adds any number of spaces to the A5 buffer. This will be used to pad out the decoded instructions as we go along. On entry, D4.W holds the number of spaces to add.

```
* ======================================================
* add_spaces :
* ======================================================
* Adds D4.W spaces to the output buffer.
*
* Expects D4.W to hold the number of spaces we wish to append.
* Expects D6.W to hold the size decoded so far.
* Expects A5.L to point to the first free space in the buffer.
* Corrupts D4 (always -1), A5 and D6 updated.
* ======================================================
add_spaces
        add.w   d4,d6           ; New size of text
        bra.s   add_next        ; Skip the first move
add_space
        move.b  #' ',(a5)+      ; Add a single space
add_next
        dbra    d4,add_space    ; And the rest
        rts                     ; Exit with D0 = -1
```

The routine is very simple, it adds the number of spaces to D6 as we are keeping a running total of the instruction, and then enters a DBRA loop to add a space to the buffer as many times as it has to. A nice simple bit of code to finish this file on.

DISS_ASM

We now have to build the decoding table. Using the information in previous articles, and the data from the last one, we add the following to the end of the DISS_ASM file.

This is probably the biggest (and most boring) change we will have to do - this issue, we define two of the big tables of data in the program. First up is the 'MASKS' table which is used when we have an op-code to decode. We mask the op-

code with the first word in each entry and check that it equals the second word.
If it dows, we have the correct instruction. The next two words give the instruction type and a pointer to its textual representation.

```
* ========================================================
* Masks table
* ========================================================
* This table is the central part of the disassembler.
* There is a single entry for each instruction and the format is:
*
* mask - word - The mask value which is ANDed with the op
* code to give ..
* result - word - The expected result of ANDing the op code
* with the mask
* type - word - the instruction type (see text)
* offset - word - Offset into op_table for text of instruction.
* ========================================================
masks       equ *
type_0      dc.w    $ffff,$4afc,0,t_illegal-op_table
            dc.w    $ffff,$4e70,0,t_reset-op_table
            dc.w    $ffff,$4e71,0,t_nop-op_table
            dc.w    $ffff,$4e73,0,t_rte-op_table
            dc.w    $ffff,$4e75,0,t_rts-op_table
            dc.w    $ffff,$4e76,0,t_trapv-op_table
            dc.w    $ffff,$4e77,0,t_rtr-op_table

type_1      dc.w    $ffff,$023c,1,t_andi_ccr-op_table
            dc.w    $ffff,$0a3c,1,t_eori_ccr-op_table
            dc.w    $ffff,$003c,1,t_ori_ccr-op_table

type_2      dc.w    $ffff,$027c,2,t_andi_sr-op_table
            dc.w    $ffff,$0a7c,2,t_eori_sr-op_table
            dc.w    $ffff,$007c,2,t_ori_sr-op_table
            dc.w    $ffff,$4e72,2,t_stop-op_table

type_3      dc.w    $fff8,$4e58,3,t_unlk-op_table
            dc.w    $fff8,$4840,3,t_swap-op_table

type_4      dc.w    $fff0,$4e40,4,t_trap-op_table

type_5      dc.w    $fff8,$4e50,5,t_link-op_table

type_6      dc.w    $f0f8,$50c8,6,t_db-op_table

type_7      dc.w    $ff00,$6100,7,t_bsr-op_table

type_8      dc.w    $f000,$6000,8,t_b-op_table

type_9      dc.w    $ffb8,$4880,9,t_ext-op_table

type_10     dc.w    $f100,$7000,10,t_moveq-op_table

type_11     dc.w    $f1f0,$c100,11,t_abcd-op_table
            dc.w    $f1f0,$8100,11,t_sbcd-op_table

type_12     dc.w    $fff0,$4e50,12,t_move_usp-op_table

type_13     dc.w    $f138,$b108,13,t_cmpm-op_table

type_14     dc.w    $f100,$c100,14,t_exg-op_table

type_15     dc.w    $f008,$0008,15,t_movep-op_table

type_16     dc.w    $f000,$e000,16,t_type_16-op_table

type_17     dc.w    $ffc0,$40c0,17,t_move_sr-op_table
            dc.w    $ffc0,$42c0,17,t_move_ccr-op_table
            dc.w    $ffc0,$44c0,17,t_move-op_table
            dc.w    $ffc0,$46c0,17,t_move-op_table
            dc.w    $ffc0,$4800,17,t_nbcd-op_table
            dc.w    $ffc0,$4840,17,t_pea-op_table
            dc.w    $ffc0,$4ac0,17,t_tas-op_table
            dc.w    $ffc0,$4ec0,17,t_jmp-op_table
            dc.w    $ffc0,$4e80,17,t_jsr-op_table

type_18     dc.w    $ff00,$0000,18,t_ori-op_table
            dc.w    $ff00,$0200,18,t_andi-op_table
            dc.w    $ff00,$0400,18,t_subi-op_table
            dc.w    $ff00,$0600,18,t_addi-op_table
            dc.w    $ff00,$4200,18,t_clr-op_table
            dc.w    $ff00,$0c00,18,t_cmpi-op_table
            dc.w    $ff00,$0a00,18,t_eori-op_table
            dc.w    $ff00,$4400,18,t_neg-op_table
            dc.w    $ff00,$4000,18,t_negx-op_table
            dc.w    $ff00,$4600,18,t_not-op_table
            dc.w    $ff00,$4a00,18,t_tst-op_table

type_19     dc.w    $f100,$5000,19,t_addq-op_table
            dc.w    $f100,$5100,19,t_subq-op_table

type_20     dc.w    $ff00,$0800,20,t_b-op_table

type_21     dc.w    $f100,$0100,21,t_b-op_table

type_22     dc.w    $f1c0,$4180,22,t_chk-op_table
            dc.w    $f1c0,$81c0,22,t_divs-op_table
            dc.w    $f1c0,$80c0,22,t_divu-op_table
            dc.w    $f1c0,$c1c0,22,t_muls-op_table
            dc.w    $f1c0,$c0c0,22,t_mulu-op_table

type_23     dc.w    $f1c0,$41c0,23,t_lea-op_table

type_24     dc.w    $f000,$d000,24,t_add-op_table
            dc.w    $f000,$c000,24,t_and-op_table
            dc.w    $f000,$8000,24,t_or-op_table
            dc.w    $f000,$9000,24,t_sub-op_table

type_25     dc.w    $f000,$b000,25,t_type_25-op_table

type_26     dc.w    $f0c0,$50c0,26,t_s-op_table

type_27     dc.w    $fb80,$4880,27,t_movem-op_table

type_28     dc.w    $c1c0,$0040,28,t_movea-op_table

type_29     dc.w    $c000,$0000,29,t_move_dot-op_table

type_30     dc.w    $f130,$d100,11,t_addx-op_table
            dc.w    $f130,$9100,11,t_subx-op_table

type_31     dc.w    $0000,$0000,30,t_dc_w-op_table
```

That is the end of the masks table. Following on we have the op-codes table. This is called 'OP_TABLE' and holds a textual representation of the individual instructions.

Take care to get the spaces and dots in the correct places. If you are confused, the value in the 'DC.W' above the text is the number of characters in the text. Hopefully, the conversion of this file from my text into QL Today's print shouldn't corrupt the layout!

```
* ======================================================
* Op_table
* ======================================================
* This table holds the text for each instruction to be decoded.
* There is one entry corresponding to an entry in the above table
* but some entries may be duplicated here or two entries above
* may point to a single entry here. The format is:
*
* Size - word
* text - bytes
* ======================================================
op_table    equ *
t_illegal   dc.w    7
            dc.b    'ILLEGAL'
t_reset     dc.w    5
            dc.b    'RESET'
t_nop       dc.w    3
            dc.b    'NOP'
t_rte       dc.w    3
            dc.b    'RTE'
t_rts       dc.w    3
            dc.b    'RTS'
t_trapv     dc.w    5
            dc.b    'TRAPV'
t_rtr       dc.w    3
            dc.b    'RTR'

t_andi_ccr  dc.w    7
            dc.b    'ANDI #$'
t_eori_ccr  dc.w    7
            dc.b    'EORI #$'
t_ori_ccr   dc.w    6
            dc.b    'ORI #$'

t_andi_sr   dc.w    7
            dc.b    'ANDI #$'
t_eori_sr   dc.w    7
            dc.b    'EORI #$'
t_ori_sr    dc.w    6
            dc.b    'ORI #$'
t_stop      dc.w    7
            dc.b    'STOP #$'

t_unlk      dc.w    6
            dc.b    'UNLK A'
t_swap      dc.w    6
            dc.b    'SWAP D'

t_trap      dc.w    6
            dc.b    'TRAP #'

t_link      dc.w    6
            dc.b    'LINK A'

t_db        dc.w    2
            dc.b    'DB'

t_bsr       dc.w    3
            dc.b    'BSR'

t_b         dc.w    1
            dc.b    'B'  ; Bcc and BRA, BTST, BCHG, BCLR & BSET

t_ext       dc.w    4
            dc.b    'EXT.'

t_moveq     dc.w    8
            dc.b    'MOVEQ #$'

t_abcd      dc.w    5
            dc.b    'ABCD '
```

```
t_sbcd      dc.w    5
            dc.b    'SBCD '

t_move_usp  dc.w    5
            dc.b    'MOVE '

t_cmpm      dc.w    5
            dc.b    'CMPM.'

t_exg       dc.w    4
            dc.b    'EXG '

t_movep     dc.w    6
            dc.b    'MOVEP.'

t_type_16   dc.w    0       ; No text for type 16!

t_move_sr   dc.w    8
            dc.b    'MOVE SR,'
t_move_ccr  dc.w    9
            dc.b    'MOVE CCR,'
t_move      dc.w    4
            dc.b    'MOVE'
t_nbcd      dc.w    4
            dc.b    'NBCD'
t_pea       dc.w    3
            dc.b    'PEA'
t_tas       dc.w    3
            dc.b    'TAS'
t_jmp       dc.w    3
            dc.b    'JMP'
t_jsr       dc.w    3
            dc.b    'JSR'

t_ori       dc.w    4
            dc.b    'ORI.'
t_andi      dc.w    5
            dc.b    'ANDI.'
t_subi      dc.w    5
            dc.b    'SUBI.'
t_addi      dc.w    5
            dc.b    'ADDI.'
t_clr       dc.w    4
            dc.b    'CLR.'
t_cmpi      dc.w    5
            dc.b    'CMPI.'
t_eori      dc.w    5
            dc.b    'EORI.'
t_neg       dc.w    4
            dc.b    'NEG.'
t_negx      dc.w    5
            dc.b    'NEGX.'
t_not       dc.w    4
            dc.b    'NOT.'
t_tst       dc.w    4
            dc.b    'TST.'

t_addq      dc.w    5
            dc.b    'ADDQ.'
t_subq      dc.w    5
            dc.b    'SUBQ.'

t_chk       dc.w    4
            dc.b    'CHK '
t_divs      dc.w    5
            dc.b    'DIVS '
t_divu      dc.w    5
            dc.b    'DIVU '
t_muls      dc.w    5
            dc.b    'MULS '
t_mulu      dc.w    5
            dc.b    'MULU '

t_lea       dc.w    4
            dc.b    'LEA '

t_add       dc.w    3
            dc.b    'ADD'
t_and       dc.w    3
            dc.b    'AND'
t_or        dc.w    2
            dc.b    'OR'
t_sub       dc.w    3
            dc.b    'SUB'

t_type_25   dc.w    0       ; No text for type 25!

t_s         dc.w    1
            dc.b    'S'

t_movem     dc.w    6
            dc.b    'MOVEM.'

t_movea     dc.w    6
            dc.b    'MOVEA.'

t_move_dot  dc.w    5
            dc.b    'MOVE.'

t_addx      dc.w    5
            dc.b    'ADDX.'
t_subx      dc.w    5
            dc.b    'SUBX.'

t_dc_w      dc.w    5
            dc.b    'DC.W '
```

And that is about it for this session. Next issue, we are back to the tutorial again where we will take a peek at QDOS and discuss extending SuperBasic with new procedures and functions. See you then.

Gee Graphics! (On the QL?) - part 15

Lines and Planes in 3 space

Herb Schaaf

In the Jan-Feb 1999 issue of QL Today there was a discussion of 3D space coordinates in GG#8 and the cover disk that came with the Mar-Apr 1999 issue had a program FIT9c5_bas referred to in GG#9 with some FuNctions and PROcedures for Lines and Planes in 3D. As I will refer to FIT9c5_bas in this article I invite you to find the disk, unzip the files and load it.

Directions in 3 space

Several terms in analytical geometry begin with direction. One is 'direction-number', another is 'direction-cosine' and yet ano-

ther is 'direction-angle'. All are used to describe and define lines and planes in 3 dimensions.

Lines in 3D space

We can set a point in 3 space as a triplet of numbers for the x, y, and z coordinates. We can then set other points with other triplets. For example we could have a first point as (3, 5, 7) and a second point as (6, 9, 19). A 3D line would passes through both points and a 3D line segment connects the two points.

Direction numbers: a, b, and c

These are the signed distances along the x, y, and z directions respectively between any two points on a line in 3 space. Let d be the distance between the two points. Since

d*d = a*a + b*b + c*c,

d is equal to the square root of the sum of the squares of a, b and c.

In the example given the direction numbers from the first point to the second point are 3, 4, and 12. The direction numbers from the second point to the first point would be -3, -4, and -12. The distance between the two points would be the square root of (9 + 16 + 144), so d = 13.

Direction cosines: lambda, mu, and nu (aka l, m, and n)

For the example given:

lambda=a/d, mu=b/d, and nu=c/d.

or

l=3/13, m=4/13, and n=12/13.

The direction cosines for the x-axis are 0, 1, and 1. The direction cosines for the y-axis are 1, 0, and 1. The direction cosines for the z-axis are 1, 1, and 0. Parallel lines in space have the same direction cosines.

Direction angles: alpha, beta, and gamma

For a line in 3 space these are the angles between the line and the x, y and z axes respectively.

For the example given the direction angles are: ACOS(3/13), ACOS(4/13), and ACOS(12/13). So alpha = 76.7, beta = 72.1, and gamma = 22.6 degrees; or alpha = 1.34, beta = 1.26, and gamma = 0.39 radians.

The direction angles for the x-axis are 0, 90, 90 degrees. The direction angles for the y-axis are 90, 0, 90 degrees. The direction angles for the z-axis are 90, 90, 0 degrees.

PROCedure line_n3space

In FIT9c5_bas take a look at lines 25130 to 25400, thank goodness for the REMarks! To try it out DIM one_line(2,3), an array of 2 points each having a triplet (x,y,z) of values. Assign values of x,y, and z for points 1 and 2 of the array.

Then enter "line_n3space one_line(1), one_line(2)". The QL will work out the values which you can examine by entering "list_array AL3S".

PROCedure parameterize

In FIT9c5_bas take a look at lines 15700 to 15580. To try it out DIM param_line(1,3,3), an array with 1 line, 2 points each (plus a 3rd register for the direction cosines), and a triplet(x,y,z) of values. Assign values of x,y, and z for points 1 and 2 into the array. Then enter "parameterize param_line, 1". Examine the values by entering "list_array param_line".

Two lines in 3 space

If they meet we can find the point of intersection, but in 3D they may be "skew lines" and not meet, however we can find where they come closest to each other. If they are parallel they could be identical lines or lines that will never meet in the Euclidean Cartesian system.

PROCedure linesN3space

In FIT9c5_bas take a look at lines 13860 to 14780, something done in 1996; thank goodness for the REMarks! To exercise the procedure RUN (or GOTO) 26420 and enter the values as requested. After entering the last value, the QL will work out the situation. You can then enter as a command line "list_array L3S" to see the array of values. I should try to improve this PROCedure; when the two lines are parallel I should find the distance and direction from one line to the other, and the common plane if the lines are not identical.

Planes in 3D space

One form for describing the plane is

A*x + B*y + C*z + D = 0

Another way to locate or "fix" a plane in 3 space is by defining its normal and distance from the origin. The normal is a line perpendicular to the plane. Since the normal is a line it can be described by its direction cosines. The origin is where

x = y = z = 0,

and we can use p as the distance from the origin to the plane.

Given 3 non-colinear points in a plane we can determine the equation of the plane, and parameterise the normal to the plane.

Plane_frm3pts

In FIT9c5_bas take a look at lines 14820 to 15660, and lines 24450 to 24480, something else done in 1996; again thank goodness for the REMarks! In line 24760 change 17520 and 17580 to 24670 and 24720. Insert line 24865 PAUSE. To exercise the procedure RUN (or GOTO) 24460. The QL will work out the values and pause between the DATA sets. You can enter as a command line

"list_array EqnPlane" to see the array of values.

Point to Plane?
In FIT9c5_bas take a look at lines 16510 to 16600 To exercise the FuNction enter the following:
"pt2pl = Pt_to_plane(0,0,0,1,1,1,-1)"

Then "PRINT pt2pl" or "PRINT PTP" will show the distance. In truth the answer seems to be the distance from the plane to the point, rather than from the point to the plane, so a change of sign is in order. The things you find when you revisit ancient writings!

Things I haven't done yet, but hope to do
Given two parallel lines in 3 space, determine if they are identical, if not identical then find the common plane and distance and direction from one line to the other.
intersection of a line and a plane in 3D
1) the line might be in the plane (and thus parallel to it)
2) the line might be parallel to but out of the plane
3) the line intersects the plane

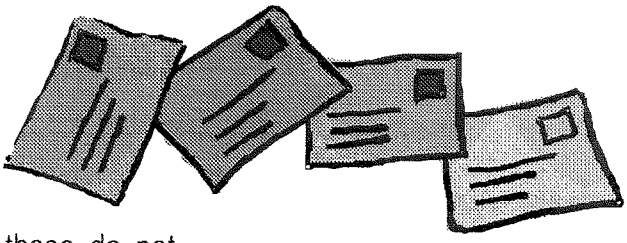
intersection of two planes in 3D
1) could be the same plane
2) could be parallel and not meet
3) meet in a line

If anyone has already worked out any of these, why not show and tell the rest of us how to do it?

Steve Poole responded to the "cutting corners" of GG#14 by sending his "CONICS_BAS", based on a program which has been in a collection of BASIC DEMO'S from the QUANTA library on DISK MD 01 since May 1993. His program draws a smooth connecting curve THROUGH all of the random points in a collection. He also sent a program "BIKE_BAS" using the algorithm to create "Chris Boardman", a neat graphic of a racing cyclist. I hope he will be kind enough to explain the logic behind the algorithm and the thought process he went through in developing it back in 1985 and 1992.

Next time?

Letter-Box



Stephen Poole writes:
Dear Dilwyn,
Thank you for QLToday volume 4, issue 5. I am beginning to regret that there is not a cover disk with every issue. Herb Schaaf's "Gee Graphics" articles are still as lengthy as ever, and it will take a lot of patience if I decide to wait for the next disk, or type in his listing for that matter. Precisely, in part 14, he asks us to consider a zigzagging collection of connected lines with their vertices, and hopes to get segments beginning to look more like a smooth curve. Well about ten years ago I needed the same sort of routine to draw contours on maps, and not finding any on offer, I began to tackle the problem myself. Some magazines showed techniques for tracing simple splines and Bezier curves, but these do not pass through the vertices.
Finally I developped a program which satisfied my needs, and sent a detailed and commented listing to Quanta for the library. The program, CONIC_bas is quite long as it first calculates all the relations between the vertices, then from perspective computes the elliptic curves in all possible sectors. There are two minor bugs which I can fix, but they only show up once in a blue moon. Some of the curves may seem a little pointed, but a close examination of the vertices shows that this is necessary.
There is enclosed a copy of the CONIC_bas program that Herb desires, with a modified version BIKE_bas to demonstrate Chris Boardman in filled vector graphics. Please could you forward all this to Herb as I do not have his address. The listings are crammed pretty tight as I kept the number of lines as short as possible during the development stage to help memorisation. At the moment I am considering a complete rewrite as, as such, they are slightly indigestible.
Regarding Q40 Colour Drivers: Will they become available to SMSQ/E SuperGoldCard users as has been suggested? An article on QL monitors would be opportune, as there has been little detail on this for a long time.
Several long letters arrived by email concerning QL MIDI, suggesting I write a program to use QL BEEPs. This has already been done and sent to Quanta. Probably I shall get in

TF Services

superHermes

A major hardware upgrade for the QL

All Hermes features (working ser1/2, independent baud rates/de-bounced keyboard/keyclick) PLUS full 19200 throughput on ser1/ser2 not affected by sound IBM AT keyboard interface (plus foreign drivers) // HIGH SPEED RS232 industry standard two-way serial port. 4800cps throughput (supergoldcard - qtpi - zmodem) at 57600bps // THREE low speed RS232 inputs (1200 to 30bps)- incl SERIAL MOUSE driver. Other uses include RTTY/graphics tablet etc // 3 spare I/O lines (logic) with GND/+5V // Capslock/ scrollock LED/Turbo/keylock connectors // 1.5k user data permanently in EEPROM

All this on a professional board about twice the size of the 8049 co-processor it replaces

Cost (including manual/software). **£90** (£92/£87/£90)
IBM AT UK layout Keyboard..... **£22** (£24/£23/£27)
Serial mouse ..... **£11** (£13/£12/£14)
Capslock/scrollock LED ..... **£1** (£1.50/£1/£1.50)
Keyboard or mouse lead ..... **£3** (£3.50/£3/£3.50)
High speed serial (ser3) lead ..... **£4** (£4.50/£4/£4.50)

superHermes LITE

All Hermes features (see above) + an IBM AT keyboard interface only. Entry level superHermes.
Cost (incl keyboard lead)...**£53** (£55.50/£51/£53.50)

Minerva

MINERVA RTC (MKII) + battery for 256 bytes ram. CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

First upgrade free. Otherwise send £3 (+£5 for manual if requd). Send disk plus SAE or two IRCs

MKI...£40 (£41/£40/£43) **MKII...£65** (£66/£63/£67)

QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

£27 incl 6 month guarantee

QL RomDisq

Up to 8 mbyte of flash memory for the QL

A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card, and reads at some 2 mbytes per second. Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off **RomDisq** at hard disk speed with only a memory expansion needed.

2 mbytes RomDisq..... **£39** (£41£37/£40)
4mbytes RomDisq..... **£65**(£66/£63/£67)
8 mbytes RomDisq..... **£98** (£100/£95/£99)
Aurora adaptor..... **£3** (£3.50/£3/£4)

MPLANE

A low profile powered backplane with ROM port

A three expansion backplane with ROM port included for RomDisq etc. Aurora can be fitted in notebook case and powered off single 5V rail - contact QBranch for details. Two boards (eg Aurora and Gold Card/Super Gold Card/Goldfire fixed to base. Suitable for Aurora (ROM accessible from outside) & QL motherboard in tower case. Specify ROM facing IN towards boards, or OUT towards back of case.

Cost..... £34 (£36/£33/£35)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface **16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)**
2 amp (for 8 relays, small motors)..... £40 (£43/£38/£44)
4 amp total (for motors etc) ..... **£45** (£48/£43/£50)
Relays (8 3a 12v 2-way mains relays (needs 2a power driver)..... **£25** (£28/£23/£27)
Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required..**£25** (£28/£23/£27)
Analogue Interface **Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temp measurements, sound sampling (to 5 KHz), x/y plotting** ..... **£30** (£31.50/£29/£30)
Temp probe (-40°C to +125°C)..... **£10** (£10.50/£10/£11)
Connector for four temp probes..... **£10** (£10.50/£10/£11)
Data sheets..... **£2** (£2.50/£2/£3)

QL SPARES

Keyboard membrane..... **£24** (£25/£24/£27)
1377 PAL..... **£3** (£3.50/£3/£4)
Circuit diagrams..... **£3** (£3.50/£3/£4)
68008 cpu or 8049 IPC ..... **£8** (£8.50/£7.50/£9)
8301/8302 or JM ROM or serial lead..**£10** (£11.50/£10/£11)
Power supply (sea mail overseas)..... **£12** (£17/£16/£21)

Prices include postage and packing (Airmail where applicable). Prices are: UK (EC,/Europe outside EC/Rest of world). Payment by cheque drawn on bank with UK address, debit card/Mastercard/Access/Eurocard/postal order or CASH! (No Eurocheques). SAE or IRC for full list and details

26 Sep 99

VISA

29 Longfield Road, TRING, Herts, HP23 4DG
Tel: 01442-828254 Fax/BBS: 01442-828255
tony@firshman.demon.co.uk http://www.firshman.demon.co.uk

MasterCard

touch directly with Al Boehm after the february holidays, and send a disk with example music, but my program can only handle duo's, not full orchestras.
For CueDark, I have dug deep into my old microdrive stock of programs and found one or two routines which may help people twiddle their thumbs. The programs are Snowflake_bas, Text_Leap_bas and Hash_bas. It is regrettable that SMSQ cannot access microdrives, as this precludes autoloading it at start-up. I have 70 tapes which I still need to access at regular intervals. Instead of copying all my tapes to disk, I have copied my microdrive catalogue to disk, as this runs smoother.

Reply from Jochen regarding colour drivers: the colours do not depend entirely on having a (Super)GoldCard, you have to have an Aurora. Roy Wood has loaned Tony Tebby an Aurora system to produce the colour drivers for the Aurora, so hopefully we're looking forward to them pretty soon.

Stephen Poole also writes:
[..] Another point I found confusing was in Al Boehm's article on Midi Sounds. He said that "Simon Goodwin's keywords via the QL NET port could handle a full orchestra"
I take this to mean that the sound is not produced on a QL, as further on he indicates that the QL can handle a chord of three 64th notes played together.
Then he states that the QL can resolve time only down to a video frame which is 1/50 second. This is quite evidently false as Beep durations can be much shorter, and rests can be programmed as short as required by using FOR loops. I have bought a book on MIDI programming and am very surprised that Simon Goodwin has been able to convert QL BEEPs to MIDI format at all. More news please.

Simon N. Goodwin replies:
MIDI and via the QL NET port implies the QL is controlling not generating the sound.
BEEP can indeed resolve shorter priods but the QL must poll the IPC (a time-consuming process) and is at the mercy of IPC (or emulation) variations. FOR loops are hopeless for timing, depending on program size, CPU, ROM version, other tasks - this is a multi-tasking system and QL MIDI keeps it that way. Indeed it's possible to use Xchange and play MIDI - in time! - simultaneously.
BEEP is nothing to do with it. Nowhere did we connect BEEP and MIDI. That is Steve's invention, mot ours, and the basis of his misunderstanding.

Al Boehm replies:
MIDI (Musical Instrument Digital Interface) is, per se, just a hardware and format standard. It specifies the hardware connections - plugs, pins, cable, etc. - that are to be used between digital musical instruments and their associated components which in our case is a QL computer. In a stroke of genius, Simon thought out a way that the QL NET port could emulate the MIDI hardware standard. Thus a simple plug can be made to connect the QL NET port to a Digital Musical instrument such as musical keyboard so that the QL can control the musical keyboard and make it play. You are right in your assumption that the QL does not make any sound; it just sends data to the music keyboard which generates the sound. MIDI can be used to send data to other devices that use the MIDI standard. For example, boxes that control stage lighting or mixers that change the volume for sets of microphones. When MIDI is used to make music, a synthesizer (synth for short) is needed to change the MIDI data into sounds which are played on speakers or recorded. Thus, the musical keyboard I mentioned above must have a MIDI IN port to receive the data and a synth to generate the sounds. Synths are also sold separately without a keyboard. One can drive a synth with a computer or with other instruments that have special MIDI pickups. I have one of these for my guitar. Also to make things more versatile, (complicated?) there are musical keyboards that have no synth built in. These are called controllers and merely send MIDI signals through a MIDI OUT port so that some other synth (or recording device) can generate the sounds. MIDI is also a format standard. Sets of numbers sent as bytes over the MIDI hardware specify what is to be done. For example start a note, or stop a note, or use a particular instrument sound. Many books on MIDI give this format. Also a lot of it is given in the downloadable files from the NESQLUG web site under MIDI:
http://boehm.home.hiwaay.net
The boehm actually stands for my son Bill who is the webmaster, and yes there are two a's in hiwaay! Many of the combinations of numbers have been made easy to use by a large set of keywords - NOTE_ON, NOTE_OFF, MIDI_MUTE, etc. - developed by Simon. These are explained in his DIY_MIDI manual which

is also on the NESQLUG web site. MIDIPlayer reads in a SMF (Standard MIDI File) and sends the MIDI signals out of the QL NET port at the proper time. Thus we come to your second main question. You are right. The QL can resolve time to less than a 20 millisecond interval. But not reliably. The QL clock through the DATE$ function only gives time to the second. Simon's timing keywords give time to the video frame rate (1/50 second = 20 milliseconds) (1/60 second for US machines). Some tricks - FOR loops - can be used to estimate time to a smaller interval. However, in a multi-tasking machine, they are not completely reliable, plus they take compute time. I worried about this quite a bit until I listened to MIDIPlayer play many songs. Now I am satisfied that it is not a significant problem. The proof of the pudding is in the eating. Try MIDIPlayer yourself. All the software is freeware or cardware (send a postcard to Simon Goodwin). You will need a synth, probably a musical keyboard, with a MIDI IN port. One with the GENERAL MIDI (GM) logo is preferred although earlier MIDI will work. Roland GS and Yamaha XG are extensions of General MIDI and play GM without any trouble. Make sure it has a MIDI IN port! Simon bought an old keyboard without a manual at a garage sale. It was invaluable for debugging the DIY_MIDI keywords, but its limitations are quite noticeable in the type of music that can be played. You will need to make a MIDI/NET port cable. Directions are on the NESQLUG web site. It only requires two wires to be soldered - no chips or resisters or anything like that. As long as you know which end of a soldering iron to pick up you can do it yourself. If you have zero skill at soldering, Tony Firshman will make you one for a small fee. The best part is that even with minimal programming skill, you can generate some neat sounds. The source code for MIDIPlayer as well as for the DIY_MIDI keywords is on the NESQLUG site. Both are loaded with comments. I have a Yamaha PSR-510, a mid level musical keyboard synth which is GENERAL MIDI. With it, can MIDIPlayer play a full orchestra? Yes, it can, and it is very sweet.

The DIY_MIDI has nothing to do with BEEP. However, your mention of it has caused me to think that MIDIPlayer can read in SMF which are available for many songs. Instead of MIDIPlayer sending MIDI out, it could send all note starts and stops to BEEP. BEEP would not do justice to many songs but something like "Flight of the BumbleBee" which is all one instrument, could sound very interesting when played just by BEEP. Let me know if you would like to work on such a project. I'll help you.

Time to Decide - Re-visited

Robin Barker

There have been, to put it midly, some interesting responses to my original Time to Decide letter. Firstly, I must make clear there is no planned agenda to turn Quanta into a multiplatform group. There has been no suggestion that Quanta's current funds would be used to support other platforms, nor has there been any suggestion we are forsaking the QL. In fact, quite the reverse. I consider the survival of the QL and Quanta to be the prime motives.

There has been some criticism that I am only interested in the future of Quanta for Quanta's sake. This I consider to be a little odd. I am after all, the current chairman and as such must surely seek and promote the interests of Quanta membership majority. The way to seek opinions is through the magazine since not all Quanta members are on the Net, hence the Time to Decide letter. The letter was purposely designed to kick-start a debate, the objective obviously being fulfilled.

Phil Jones asked whether the committee have pursued a previous idea to increase membership that involved mailing letters to ex-members asking why they left Quanta and enticing them to rejoin. This was tried and proved to be unsuccessful.

I suspect the majority of members leaving Quanta fall into two groups. The first will have given up computers completely, for reasons that would include health etc. The second will have moved to other platforms in favour of the QL. I believe we should try to encourage this group to stay, but how? If we could maintain their interest in Quanta, therefore keeping them in touch with the QL, they may, for example, end up purchasing an emulator! It is surely better to have the QL on the shelf rather than out the window.

To pretend PC's etc., don't exist is banal. Not only do a significant number of QL'ers use other platforms as a matter of course, many QDOS based systems are now run on other platforms. It therefore follows users also have an interest in those other platforms. Do we accuse those who run QDOS type systems on a PC as being deserters of the QL? Of course not! Why not in this case, attend to their interests. I would guess as time goes on, more and more QDOS based systems will be run on other platforms, mainly the PC. Times have changed. No longer is the QL isolated in its black box. The point is, do we change?
As the QL user base continues to decrease development becomes more difficult, especially hardware which becomes increasingly expensive as demand falls, and like it or not, demand for updated QL products is the key to the QL's future, for without this, the QL will become an antiquated toy. This is the reality, so what are the options? Do we attend to members interests in other platforms via the magazine? (For those who would not wish to read any such articles, you simply skip the page just like any other magazine you read) Do we invite other platform users into the fold so they at the very least, become acquainted with the QL? or what? You tell me.
The QL 2000 show in October will feature a debate on this subject on the Sunday morning. The outcome of that debate, taken together with your correspondence, will be crucial. You are the Members. You Decide.

Robin Barker
Chairman, Quanta

Do you have comments, questions, other kinds of feedback? Just write to us and we will try to help or contact the authors. Stephens letter has inspired Al, and the result of his feedback comes next:

MIDIPlayer 2

Al Boehm

After much testing, delays, and additions, MIDIPlayer 1 went out on 8 December 1999. The next day I thought I would download some MIDI Christmas music for the season. I found some good ones. loaded them up, but they did not work!
A good thing I had MIDIView. This a SBasic program I wrote to decipher SMF (Standard MIDI Files). SMF is the standard format for storing MIDI data on disk. Music stores sell MIDI songs on disk or CD and when MIDI songs are downloaded from the web, they are in SMF format. MIDIView reads a SMF and prints what it interprets each MIDI event to be.
In less than a minute, I saw what the problem was. In SMF, MIDI events generally consist of three parts: a time, a status byte, and data bytes. The status byte indicates what is to be done and how many data bytes follow. There are several exceptions to the three part rule. One of the most common is what is termed "running status". This means that the status byte (which is always greater than 127) is left out. Instead use the status from the previous event on the same track.
MIDIPlayer 1 allowed for running status on SMF type 0 which is limited to one track. SMF type 1 uses multiple tracks. Usually, but not necessarily, each track is for a separate instrument. For example, track 1 for clarinet, track 2 for trumpet, track 3 for bass, track 4 for drums, etc. Now I thought who would want to run status with multiple tracks. The clarinet needs a different status than the trumpet and so on.
Well I was right in the sense that you can not run status out the NET port when instruments are intermixed. However, on the SMF file, you can save bytes by storing the events in a run status mode if the same instrument is used on a track. Thus, track 1 would have one status byte at the start to let you know it was for the channel assigned to the clarinet and from then on nothing but time and data bytes.
Of course, in sending MIDI data out the NET port, you would have to add the proper status byte anytime another instrument came in-between your last use of the clarinet.
Well a couple of IF statements could do the trick. The trouble is the QL could have trouble keeping up if the IF statements were added in the PLAY loop. It took me over two months two figure a way to check beforehand, and when needed, POKE status bytes into the time section of the file stored in ram. This is possible since SMF time (in beats per minute and fraction of a beat since last event) is recalculated into QL frames (1/60 second) and stored separately in an array. The address in ram to the MIDI event was reset to the new status byte. Since this is all done beforehand, the PLAY loop did not become a bottleneck.

Thus MIDIPlayer 2 can now run type 2 SMF even if they run status on multiple intermixed tracks without any degradation in speed. Finally I could listen to Winter Wonder Land using my QL!

If you haven't heard MIDIPlayer play, you are in for a treat. You do need a synth and several people have asked me what should they get. If a musical keyboard has the General MIDI logo it will do quite well. I have seen these in catalogs as low as $130 new. General MIDI requires 16 channels and up to 24 simultaneous voices or 16 simultaneous melody voices plus 8 for drums. However, almost any of the new musical keyboards that have a MIDI IN port have enough capability to work OK even if they are not General MIDI. But if you buy an older one at a yard sale make sure it is General MIDI or at least has a substantial number of channels and simultaneous voices.

MIDIPlayer 2 is available on the MIDI section of the NESQLUG web site

http://boehm.home.hiwaay.net

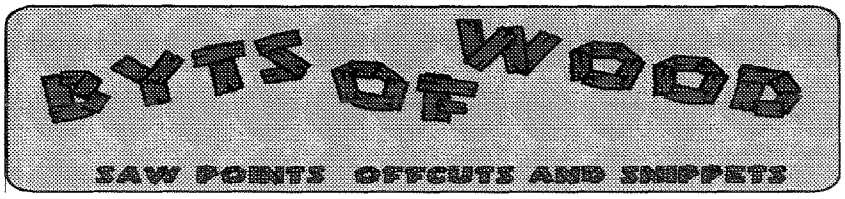
along with Simon Goodwin's DIY_MIDI keywords and documentation. Source code is available too. Also there are several MIDI documents including "An Intro to MIDI".

I have not put MIDIView on the NESQLUG web site since it needs some cleaning up and a manual before I would consider it presentable. However, it works, and I will send it to anyone who can use it.

As nice as MIDIPlayer is, I consider it a very beautiful toy. My next planned project is to beef up MIDIView into an easy to use low level MIDI editor. This capability is something I need. The high end MIDI editors for Mac, PC, etc. are available (usually expensive) but just do not have the capability to change a single byte.

Of course, I thought I would finish MIDIPlayer in six weeks; it took nine months! So don't hold your breath!

Oh, in answering Stephen Poole's letter about MIDIPlayer and BEEP, it occurred to me that MIDIPlayer could be converted to use BEEP. Of course, BEEP would not be melodious as a synth and would have trouble with multiple parts. But a solo part could be sounded by BEEP. Is anyone interested in working on this project? I'd be willing to help.



A Quick announcement!

This column was produced entirely on the Q 40 which is now the main machine in The Q Branch HQ in Portslade. The shop and the shows will still do their business through the MinisQL because it is easier to transport and my Q 40 will be there to demonstrate its potential but this is now my machine of choice for producing adverts and most QDOS /SMSQ use.

Normal Service is resumed

After the flurry of the New Year/New Century/New Millenium celebrations at the start of the year had died down we got down to some serious QDOS/SMSQ business with three shows one after the other during February. First off Tony Firshman, Bill Richardson and myself attended the Radio Rally in Cambridge, followed by an Eindhoven trek and finishing off with my own workshop in Hove.

This could be a good year for workshops with several in the planning stage. Apart from the U.S. meeting there are others planned in Paris, Austria, Italy, and Ireland as well as the QL 2000 meeting in Portsmouth in October.

If anyone is planning a show please send me the details so I can include it on my website. Those of you who are online and want to check show details can do so through the Q Branch website at:

http://www.qbranch.demon.co.uk/websites/

The Cambridge one was strangely curtailed and, having got up at some silly hour of the morning, and driven for two and a half hours to get there for 9.00am the show began packing up at 2.30pm. I apologise to anyone who arrived after that to find us gone. As it was a couple of QLers turned up after we had packed our stand away and both Tony and I had to dig out some of the stuff they wanted.

Eindhoven

This regular meeting is one of the places where some of the issues get thrashed out since it is a relaxed venue and always followed by a visit to the local Chinese Restaurant where traders, programmers and some of the customers can have a general discussion.

This meeting was enhanced by the presence of Tony Tebby and Peter and Claus Graf (thanks for the meal, guys). as well as several members of the French QL Club. Q 40s abounded running three different operating systems although mine was pretty much taken up for most of the day with our attempts to get LINUX running on it. We did make a few interesting discoveries about this (see the Q 40 section) and I did leave the venue with LINUX completely installed on my machine.

Eindhoven also saw the launch of SuQcess, Wolfgang Uhlig's front end for the DBAS database system. I originally saw this in Eindhoven last year and was impressed then so I am very glad to see it released. I am even more pleased to see a new programmer surfacing.

We need more!

Wolfgang Lenerz was also there taking notes and suggestions for improvements to his Agenda program. It is good that programmers listen to their customers in this way and make the kind of changes that are needed to improve the product. Eindhoven is a long way from Hove but always worth the journey.

Hove

My usual thanks go out to all of those who attended or helped the Hove show at the end of February. This is always a hard show for me because I have to be both trader and organiser. When we arrived we found that there was a wedding reception going on in the other room and there were not enough tables available. This is despite our having booked three months in advance and specified that we would need lots of tables. Tony Firshman and I went on a raid and stole tables from various places when the lazy hotel staff could not be bothered to help us. I think I shall be looking for a new venue for next year's show.

Enrico Tedischi was at this show displaying his collection of all of Sinclair's products (he couldn't get his C5 down the stairs) and he will be attending the big two day event in Portsmouth later this year.

It was a shame that ill health prevented Rich Mellor and Dilwyn Jones from attending the show. Rich is in bed with a nasty stomach condition which does not allow him to sit up for longer than an hour at a time. Get well soon Rich.

A Few Old Chestnuts

Many people have complained in the past about BOOT file listings but there is a place for

RWAP SOFTWARE

QL Cash Trader v3.5 £5

A well established accounts package for the small to medium sized business, including automatic generation of profit & loss account, balance sheet, VAT return, reports and analysis for audit trails and management decisions. Previously sold for over £100.

QL Payroll v3.5 £5

Manage a payroll for a small to medium sized business. Handles up to 99 employees easily, producing P45s and P60s as well as the payslips on a monthly or weekly basis. Calculates tax and national insurance and is easy to update to take account of current tax year rules.

Q-Help v1.03 £10

Provides on-screen help for SuperBASIC commands, including most toolkits (TK2, Turbo Toolkit, SMSQ/E and PD toolkits). Copy of PD toolkits available separately for £2. Can easily be used to add help pages to your own programs - simply produce ASCII text files containing each help page, an index to the help page and the program will automatically cross-reference and display the links o screen.

Sidewriter v1.08 £10

Produces landscape printouts of Easel/QSpread spreadsheets and output from QL Genealogist, as well as any other standard text file. You can specify the fonts to be used on the page. Works with all EPSON compatible printers, from 9 pin dot matrix up to inkjet printers. A most useful utility written by Dilwyn Jones - you know it must be easy to use.

ProForma ESC/P2 Drivers v1.01 £8

New improved printer drivers, providing up to 720 dpi printout in full colour from all programs written for use with ProWesS, such as LineDesign and Paragraph. Work on all Epson inkjet printers which support binary mode compression (740,850 and 900 models at least). 1440 dpi to follow.

QL Genealogist v3.24 £20
Genealogy For Windows £50

These programs enable you to keep a track of your family tree - add individuals, with details of their parents and children, and watch all of those links build up into a formal family tree layout. Notes and pictures connected with an individual may also be stored, making this the perfect way to preserve the history of your family for future generations. The QL version now allows you to keep details of both the male and female side of the family in one large family tree. Easy to use with a step by step tutorial for starters. This is a special offer price - valid until 31/3/2000.

SBASIC SuperBASIC Reference Manual £40

Updates £6 each, £10 for 2 (Current Version - Rel 3)

Have you ever tried to write a program, but been lost as to the means of performing a certain action? This Reference Manual provides you with a full description and examples of how to use all of the keywords found on a standard QL, plus the keywords under SMSQ/E, Toolkit II and many different public domain toolkits. Details of any possible problems are provided, together with descriptions of how to use the device drivers and how to ensure that your programs are compatible across the range of QL platforms.

This book is ideal for all QL users and is kept up to date by regular updates.

Orders are currently being taken for the next print run of this popular tome.

(Note Price for the book does not include postage and packing).

Return To Eden v3.08	£10
Nemesis MKII v2.02	£8
The Prawn v2.01	£8
Horrorday v3.1	£8
West v2.00	£5
The Lost Kingdom of Zkul v2.01	£5

Classic QL adventures, now re-released without any need for microdrives. These include mainly text adventures, catering for all tastes, from the spoof Prawn, through to a Hammer Horror, fighting the bad-guys in the old West and battling with robotic hoards and goblins. Return to Eden is a massive three disks of adventure, with pictures for each location and a captured prince to rescue. With three characters to control, each with their own abilities and skills, this one should keep you amused for many an evening.
All six adventures are available together for only £25.

FlightDeck v1.04 £10

Can you learn to fly a twin-engined passenger jet? This simulator includes full shaded 3d views of the world around which you are flying, toegether with the ability to add navigation beacons, airports and even landscape features to make FlightDeck the ultimate QL Flight simulator. A database of the main UK airports is included to allow you to fly around the UK for training missions.

Q-Route v1.08C £25

This is the latest version of this popular route finding program. Find the quickest route or the shortest route between any two places, using roads. A wide ramge of maps is available for this program. (see elsewhere in this advert). The program is easy and quick to use. You can even add your own places and roads to the maps to include local detail.

D-Day MKII v3.04	£10
Grey Wolf v1.8	£8

For the wargaming enthusiast - D-Day is a classic table top wargame, where you control either the Allies or the Axis forces and play against either the computer or another human player. With the ability to define your own army set ups and a choice of four different scenarios, this should keep you entertained for a while. Grey Wolf places you in charge of a submarine - can you sink the enemy shipping whilst avoiding their planes and destroyers??

A range of games to keep you amused on the QL. Some are old favourites, like Golf and a quiz program (with over 500 questions). Whilst others are fast, colourful arcade games. Plenty of variation and skill required - what more can you ask for? All 5 programs £20 only.

Open Golf v5.20	£8
QuizMaster II v2.07	£5
Stone Raider II v2.00	£5
Hoverzone v1.2	£5
Deathstrike v1.5	£5

These are the latest maps for Q-Route (now at v1.08C). Find your way around the various countries covered. South and West Yorkshire Map is a much more detailed area of that beautiful part of the British isles.

Britain.map v1.09	£2
BIG Britain Map (needs 2MB) v2.01	£5
South & West Yorkshire Map v1.03	£1
Ireland Map v1.00	£5
Belgium Map v1.01	£2
Catalonia Map v1.01	£2

RWAP Software, 4 Anvil Crescent,
Coseley, West Midlands
WV14 8GA
TEL: 01902 836888

* Also known as Trading Accounts

Cheques in £sterling payable to 'R.Mellor'

these. On the QL-USERS mail list someone posted a copy of his BOOT file because he was having a problem. As always on this list, there was no shortage of useful remarks about it. Amongst those was one which all users of Qubide and multiple partitions should note. Each partition takes up a sizeable chunk of RAM to store its map. If you are in the habit of linking all of the partitions in when you start up, think again. You will be amazed at the amount of memory saved when only one WIN1_ is active. You can always link more of them in with either a WIN_DRIVE command or Winlink.

Another point was made by Malcolm Cadman when talking about Mice and the QL. If you use some 3-button mice and think they don't work try holding the left mouse button down when firing it up. This will force the mouse to fire up as a three button mouse and not as a two button one. On my old SERmouse system this was vital in order to stop the mouse streaking off into a corner and sulking there. I have the same problem with the Aurora system in the Q Branch HQ but not with the MinisQL both of which use the full superHermes. Different mice - different techniques needed.

This Gold's on Fire

Recent communications between Nasta and us suggest that thinking about the Goldfire is reaching an advanced stage. One of his suggestions concerns the amount of RAM to be available and which form it should be in. My recent experience with the Q 40 and 'standard' PC items such as SIMMs suggests that there is no such thing as standard. the upper limit of the memory addressable by the GoldFire is 128Mb - massive by QL standards average by the megalithic PC. You could have this on a 'standard' PC 168 pin DIMM or soldered directly onto the board. Since a 128Mb DIMM refused to work with my graphics adaptor on my PC recently even though it worked fine with a different PC/Graphics Card I personally think that there is no contest. Find SDRAM which works and solder it directly onto the board. This does, however, raise one slight problem. The Q40's performance is held back slightly by the way in which the memory slave blocks are handled. Users with 32Mb as opposed to the standard 16Mb supplied find a slight delay in the performance as it checks the extended slave blocks. What would this do to a 128Mb memory I wonder?

I think we would all rather splash out a few more pounds and have a large memory capability on our systems and I think that there is no point in supplying lower memory units at a slightly lower cost. For once the capacity is not the issue here since ram chips themselves are comparatively cheap. Performance is what counts. We should, therefore look for ways around the slave block problem. Would it be feasible to specify an area and size for the slave blocks to use? These are mainly used when loading from disk so why not allow the user to specify how much ram should be allocated to slave block use. If you do a lot of shifting of data from disk you may want to increase this or, if you do not do much and tend to work on particular topics for long periods you could restrict it to a couple of Mb's. I am no programmer so I am not sure if this is a feasible idea.

Thierry was involved in the discussions about the GoldFire and, when the talk turned to the thorny problem of compatibility with older games and programs, suggested that anyone who bought the more advanced systems such as the GoldFire, QPC 2 and the Q 40 would not want to use these older programs. I think I would differ here. I have already had people call me about running Abacus and Perfection of the Q 40 and about printing from Quill or Perfection from the QPC 2 emulator. It seems to me that people will always want to use the programs that they are the most familiar with and it is interesting to see people with 500MHz PCs with all of the Micro$oft software on them wanting to use Quill. This should tell programmers everywhere something about simplicity. I suppose they are not listening though because they are so busy writing animated paper clips and other such nonsense.

All the same the problem does remain that we have to sometimes sacrifice advances in our system to take into account the way that older software exploited loopholes in the old ROMs. People tell me that SMSQ/E is faulty because it will not run some of the old software without realising that the old software only ran because either the O/S did not see the errors in it or it was writing to an area which it was not supposed to but which was not being used at the time. There were areas in the JM and JS ROMS which were reserved for future use but never got exploited. Some of the old software used these areas and now, when the newer O/S comes along and uses these areas there are conflicts. It seems to the user that SMSQ/E is at fault but that is often (not always) the case.

A recent bug fix for SMSQ/E

(v2.95) actually made the situation worse and some software crashed because of it. That should be fixed by now so upgrade immediately if you have older versions.

Teaching a New Dog Old Tricks

One of my shop customers, who runs the computer systems in a local company, called in a while ago when I was running up the first batch of the Q 40s. He wants to write his own operating system and became fascinated by the simplicity and elegance of the Q 40's design. He immediately bought one of the first units and has been working with it for a while now. He is the first user who has not come from the QDOS/SMSQ direction and this is interesting for me because it tends to highlight the differences between our system and others.
He was amazed by the concept that you could write a piece of code and add it to the system to generate new keywords and has been eagerly throwing himself into getting to grips with, what was to him, an alien system. Having been a QL user for over fifteen years I do tend to take some things for granted and I have had to think back a long way to get explanations about the way things work for us. I do not think that we will see any software from him although he is learning fast but I will keep you informed about his progress if you are interested. I may even be in a position to offer yet another O/S for the Q 40.

LINUX

As I mentioned above I have now got a functioning LINUX installation although time has been a little short recently so I have not done more than dipped my toe into the surface of it. This seems to exist on two levels and be perceived by people in two very opposite ways. One thing people seem to want more than anything else is a system which does not crash - understandable really. If you talk to the average PC user they have perceived the hype about LINUX as 'This is Windoze that Works'. Well that may be so or it may not. It is hard to tell this at this stage since I have not really got into using LINUX and there seem to be very few applications in the LINUX package so I suppose I will have to start raking the net for 68k LINUX programs.
Purists and programmers eschew the X-Windows approach and go for the command line control. Since my knowledge of this is very limited I have not investigated it to any degree but, as I have said in this column before, I agree with Stuart Honeyball who thinks that computers should be accessible and understandable with very little reading or study. I suppose it all hinges on what you have a computer for. If you enjoy playing around with the system and programming then that is all well and good but the average user just wants to get on with the things he bought the machine for. Typing out an essay at the command line is just not for him and that is why systems like Windoze took off in the first place.

HTML

There has been a lot of discussion on HTML in the QL-Users Internet group and in this issue of the magazine. I got involved in HTML a while ago when I rewrote the DATAdesign manual when PROGS released ProWesS. It seemed to me to be the ideal tool for producing documentation because of its ability to allow the user to jump around the text. I am still not 100% convinced that reading manuals onscreen is the best way to do it, however. The more involved I get with PC software whose documentation is almost exclusively in that format the more I feel that I do not get the full potential out of the programs. I would very often buy a new program and then sit down and read the manual for a while to get an idea of what it could and could not do. If you only have the manual as an online help source you will inevitably miss some of it's potential.

Adjust the Webbing

Like many QLers now I have a PC sitting on my desk. I built it so that I could play a few games and as an Internet station. In the second part of that I include maintaining my website. It is in that capacity that I come more involved with HTML. The PC abounds with programs to produce HTML There are loads of them. The abiding factor which distinguishes them is that they nearly all produce absolute rubbish HTML. Spurious tags abound and it seems that they put in stuff just to ensure that the resulting files are large enough to qualify as PC files. There are a couple of interesting ones and, amazingly enough, these are both share or freeware. NoteTabPro is shareware from Switzerland which does a good job and has lots of other uses and Evrsoft produce a free editor (1st Page 2000 - **www.evrsoft.com/1stpage/** - Steve Hall found this.) which is superb but these are PC programs. Dilwyn and Geoff Wicks are both able to convert files from word processors into HTML but these, by their very nature, must produce restricted

code. We need someone to take on the task of writing a proper HTML writing tool for QDOS/SMSQ systems. I would imagine that the resulting editor would not be an easy task because, to make it really useful it would have to be able to preview the results.

The corporate vultures of big business do things like charging a high licence fee for people who want to use GIF image rendering and these things also need a lot of colours for them to work but JPG files are now a common internet picture form and Dave Westbury proved that a JPEG viewer is a possibility so we should be able to do this. I would love to be able to produce and edit my website on a QDOS/SMSQ system rather than on a PC.

A Rolling Format Gathers Much Mess

The big problem with HTML for us QLers is that it does not stay still. The only two readers that we have for the QDOS/SMSQ system are the PROGS one, supplied with ProWesS, which will only handle HTML 3.00 or lower and QLynx - a PD HTML reader ported by Jonathan Hudson. I must confess that I have not tried to get QLynx working after a few abortive attempts a year or so ago but the whole process seemed unnecessarily complex. I do not, therefore, know if it can handle later versions of HTML. The kind of HTML that we meet on the internet today is of a much higher level and we desperately need a reader that can handle this if we are to proceed. The closeness of the release of the full colour drivers for SMSQ systems and the work being done TCP/IP stacks means that a full function HTML XTML viewer should be a priority. Which leads me to this month's .....

As you may have seen in other places in the magazine PROGS have made the ProWesS base distribution free under the GPL licence. Joachim did this to encourage someone to use his HTML viewer, provided with it, as a base to produce a more up to date version. This does not mean that PROGS are quitting the QDOS/SMSQ world just that they are making their stuff more accessible which can only be a good thing. It requires a 'C' programmer and someone with a good knowledge of HTML to do it but we could wind up with a web browser. One of the main accusations levelled at ProWesS is that is a slow in comparison to other programs on our system but this is really a result using vector drawing techniques to produce the screens. As we move more towards faster machines such as the Q 40, QPC 2 on fast PCs and the GoldFire ProWesS should come more and more into the picture. Most people who move up to more powerful systems have a problem when using the higher resolutions because the size of the characters makes it hard to read. Only programs like ProWesS can overcome this because the characters produced by these programs are vectored and not pixel based. We are planning to distribute the ProWesS base installation as a free cover disk in a later QL Today. Look out for it.

POSTCARD: Screen Saver - You decide!

You will find a little postcard with this issue. Please answer the questions on it and post it back to Jochen Merz Software.

If you have used the CueDark screen saver system presented on the last cover disk, please take a few moments to look at the modules presented on this disk and vote for your favourite module to help us decide on a winner! Note that Thierry Gidefroy has asked that his modules are not entered into the competition, so please do not vote for him even if you preferred his modules!

The card also asks you to tell us if your system can read HD disks, which will help us since it is becoming harder for us to obtain DD disks now.

The final section lets you tell us what sorts of cover disk "themes" you'd like to see in the future.

Please enter your name and email address. There is a small box you can tick if you would like to be added to the QL News Mailing List so that we can keep you informed about news and shows, especially QL shows in your area.

US QL 2000 Show

Sponsored by NESQLUG

White River Junction, Vermont USA on May 20, 2000

The show will be held at the Hotel Coolidge in White River Junction, Vermont USA on May 20, 2000. This small town is right on the border of the states of New Hampshire and Vermont in the heart of the US region known as New England. Close to major Interstate Highways I-89 and I-91.

The Hotel Coolidge has a web site **http://www.hotelcoolidge.com/** and can be contacted by email (**hotel.coolidge@valley.net**). There is also an 800 number that will work within the US and Canada (800-622-1124) and a regular number (802-295-3118). If you call please mention NESQLUG or QL Computer Show to get special rates:
Single $49 Double $59
I have reserved 8 rooms to be held until April 20th but will reserve more if we need them. All rooms are non smoking. The hotel has a restaurant and bar. For those interested in the experience or on a tight budget will welcome travelers to stay in our home. Just let me know.

As in past years on Friday evening, May 19, at 6 PM we will meet in the hotel lobby and go to dinner together. The show will be at the hotel from 10 AM until 4 PM on Saturday, May 20. A lunch will be served during the show. Admission to the show including lunch will be $12 per person but traders will be free. After the show there will be a dinner at the hotel which will be $20 per person. On the following day, May 21, there will be an all day gathering at our home, Bill Cable/Mary Boyle, for the day. It is located in Cornish, New Hampshire which is about 15 miles from the Hotel Coolidge. See the Wood And Wind Wind Generator in action.

Please contact: Bill Cable Director of NESQLUG at **cable@cyberportal.net** or (603) 675-2218 if you have any questions. The Hotel Coolidge Web Site has a map and more detailed maps and directions will soon be available at the NESQLUG Web Site. If you are planning to come please let me know so we can keep an accurate list of who will be there. Hope to see you there.

Airport	Driving Time to Show Location	
Boston, MA Airport	3 hour drive north	
Manchester, NH Airport	1 1/2 hour drive north	Nice Drive
Hartford, CT Airport	3 hour drive northeast	
Burlington, VT Airport	2 hour drive east	Nice Drive
Montreal, Canada Airport	4 hour drive south	Nice Drive
New York, NY Airports	6 hour drive northeast	
Lebanon, NH Airport	10 minutes drive	
this is small regional airport has rental cars		

The QL Show Agenda

16. April - Davyhulme, United Kingdom
NEMQLUG Workshop & Quanta AGM

13. May Eindhoven, The Netherlands
St. Joris College, same venue as always.

20. May East Coast US Show. See reverse side!

Planned for September: Italian QL Show.

Planned beginning of October: Austrian QL Show.

14./15. October: QL 2000 Show (see below).

QL 2000

Roy Brereton informs us, that the Quanta sponsored big two day QL2000 workshop style meeting is set for Saturday and Sunday, October 14 and 15. The venue is:
The Horizon Center, Sundridge Close, Cosham, Portsmouth, PO6 3LP.
Cosham is a northern suburb of Portsmouth. Roy is the Quanta point-of-contact. At this moment he is sorting out the various hotels to find the best deals. Overseas groups are being especially invited to come to this show. Saturday will be a trading day with several QL notables making presentations. Evening will see a group dinner, subsidized for Quanta members. On Sunday at 10:00 AM there will be a debate to discuss the future of Quanta. At noon normal workshop activities/trading will continue.
At least 16 people from the US have expressed interest in coming. NESQLUG has appointed Al Boehm as point-of-contact. Al is collecting travel information and any potential attendee from North America is encouraged to contact him: boehm@ziplink.net
or Mail: 2501 Ermine Dr., Huntsville, Alabama 35810, USA.
"Everything" about Portsmouth can be found at:
http:/www.portsmouth-college.ac.uk/johnsite/Local.htm